ADVENTURE

GROUND COMBAT

The ground combat system is a single system that will be used for both melee and ranged combat. It is RPG-oriented and will be largely automated to allow for a high-degree of lag-tolerance. This system will also give players a significant level of control and ability to tune what their character is doing.

The combat system for Star Wars Online is RPG-oriented. It is *not* a quake-style first-person shooter where you run around with a keyboard and a mouse while left-click rail-gunning targets over your broad-band connection using Gamespy to find a SWO server with a ping of less than 100. The key ingredient in a successful combat encounter is luck, skill, and cunning because it all boils down to target numbers and die-rolls modified by skill-levels.

Vision

The game will employ an advanced combat system based on that of MUDs. Each type of attack or weapon will have its own delay, permitting discontinuous "rounds." Different types of attacks can cause delays in others, change their "positions" or their focus, and affect their ability to get off the next blow. Shots or blows will occur automatically, to serve as a barrier against latency being too big a factor in combat. However, special attacks will require manual intervention, and will be queued up. The queue may be cleared at any time by a player (and may even be visible to them onscreen as icons).

Missile-based combat will be handled similarly to melee. In addition, overall modifiers such as jump moves, wary/aggressive settings, and targeted blows will also be possible. Posture (such as lying prone, kneeling, etc), terrain, and aimed shots will also add to the tactical depth of missile-based combat. In addition, secondary fires (such as stun shots), and John Woo-style moves (diving while firing) will be available.

All of this will be usable both against players and creatures, according to the PvP rules (see below). Creatures will not be spawning in static, predictable locations. The spawning mechanisms will be tied to players themselves, so that creatures are not using up server resources when the players are not present. This also allows us to spawn creatures appropriate to the players' capabilities at all times. A sense of narrative flow will be maintained by having locations on the map be more appropriate for certain types or difficulties of creature. Creatures will also improve in their capabilities and be saved in the database if they survive long enough, named uniquely and rumors of their existence automatically put into the news system.

Terms

Primary Engagement Zone

Under the basic combat model, this zone is anything in the front hemisphere of the player. This is a field of engagement where a player can shoot or swing a weapon with no penalty, as their target is more or less directly in front of them.

Under the advanced combat model, the size of this field is primarily modified by the character's current stance. It is usually close to the same as the player's first-person view would be. A target outside this zone can be engaged by swinging or shooting off to one side or the other, but this will entail a slight to moderate penalty depending on the bulkiness of the weapon, as it is more awkward.

Macro

An option under the advanced combat model, a macro is a player-constructed chain of tasks that are assembled and then saved into one of several macro- 'slots' or simply saved off under the 'macro' command sub-listing. Macro's are used as a time-saving measure of customizing a player's combat tactics, due to the player's ability to pull up a macro and dump it into their queue to be executed immediately.

System Details

Core Requirements

- Asynchronous task-based combat
- Reactive tasks
- Varying character condition states
- Basic combat model versus advanced combat model

Fights are an asynchronous task-based challenge between all involved participants. Each player has a queue of tasks that will be completed in order, and the server resolves the execution of those tasks. As each task is resolved, the next task in the cue is then undertaken.

While it will be almost impossible for a player to directly react to anything, player-modified settings will allow them some measure of control over how their character responds to attacks. In addition, some tasks will be relatively reactive to the current situation such as firing, for example, a character would cease firing at a target that has moved behind cover after several seconds have gone by.

During combat, hits and task execution will cause the character's overall conditions to change at varying rates into different states including unconsciousness, exhaustion, and stunned. In these states the character is more vulnerable that they normally are, and are generally unable to undertake any tasks, possibly ending combat. Characters can also undertake tasks that will help increase their current condition by using medical kits and such.

The Phases of Combat

Target selection

In order to attack a target directly, the game has to know what you're trying to attack. This could be one character or a group of characters. To attack a single character, the player would simply select them by clicking on their avatar or pull their name out of the list of available targets. To attack a group of characters, the game could be told to attack the closest target in the group within their line of fire, or to select a target based upon other priorities such as a player-selected priority. Some weapons may be capable of attacking several targets simultaneously assuming they are all within range. If several characters were within close range during the execution of a light-saber attack, then

several of them could be hit with a light-saber swing. They could also all be hosed down by an automatic type of firearm or a flamethrower. Then again they might all be missed, too.

The attack(s)

To actually attack a target, the player need only have a target selected and a weapon readied. If the target is within reasonable constraints for range and in sight, the attack is carried out. The attacks will continue a set number of times, or indefinitely, depending on what the player has selected.

Each attack task will vary in length depending upon the character's strength, condition, skill, stance, and what weapon they are using. Generally, a stronger, more highly skilled player will be able to yield a lighter, better made blade more quickly than a person without these traits.

In most cases, the character will repeatedly attack again and again as soon as they recover from the previous attack.

Attack Modifiers

The formula to check if an attack hit or not will need the ability to be modified on the by the server in order to react to changes in the combat situation. It will be based upon the weapon's ability to hit a certain sized target at a particular range. This target number is made more difficult or easy by additions and subtractions based upon whatever actions and defenses we see fit to track.

Differences between the basic and advanced combat models

The basic combat model covers what is essential for there to be ground-combat in the game. While it lacks some depth, the asynchronous combat system allows players to hack each other up and shoot each other while still being fairly lag-tolerant. However, by adding some additional features, we can make combat far more tactically interesting.

Player Queued Moves

There will be times that the player might want their character to do more than simply move around or pull the trigger and swing their sword over and over again. These tasks would be pulled out of a catalogue of tasks the character currently knows and inserted into the queue in the order the player would like them executed. These moves could be set to immediately pre-empt the task the character is currently working on, or to wait until the player character's current task is complete. These tasks could be extremely simple or extremely complicated. A simple task might be lying down and going to a prone position. A complicated task might a John-Woo style leap from cover with blasters blazing, or a back-roundhouse kick immediately followed by a foot sweep all done Bruce Lee style.

Cautiousness

The player has the ability to control their cautiousness. This is a scale of how much the character is preparing for each attack. It acts as a modifier on the player's normal attack sequence, affecting how much they are bracing themselves or aiming prior to each attack.

A player that has totally thrown caution to the wind and is acting extremely aggressively will have set their cautiousness to its minimum value and will be snapping off attacks as quickly as their character and weapon will allow. This means that the attacks will executing far more rapidly but will also be slightly less accurate and far more wild, as well as coming more rapidly but with less strength behind them. A player that is being more cautious and taking their time with each attack will brace themselves carefully before each swing or aim very carefully before each shot, taking more time but increasing their chances for a hit/strike significantly. This applies in slightly different ways to both ranged and melee combat.

The base target numbers for the ranged-weapon attack task will be assigned as if the character were shooting from the hip, like it were being pointed but not truly aimed. Under the default setting for cautiousness, the cautiousness modifier scale will automatically add one aim task to the beginning of each ranged-attack task. This will be the equivalent of bringing up the ranged weapon and glancing at its sites to get a rudimentary sight picture, but not actually carefully aiming.

For situations where the player can take their time, they will have cautiousness maxed and carefully aim each shot or blow. If the character is at point-blank range with an opponent, they could snap off many shots or a flurry of blows by having cautiousness at its minimum setting.

Auto-Dodge

In addition to having the ability to control how carefully or recklessly the character makes their attack, the player can also control how vigorously their character defends themselves. This modifier is, like cautiousness, controlled through a sliding scale ranging from a completely defensive posture where all the character's behaviors are organized around avoiding and blocking blows, to a setting where the character has ice water in their veigns and will not bat an eye while under attack. If a dodge *is* triggered, it will mean the player will drop whatever their current task is in order to execute the dodge or parry task as well as whatever additional actions their current auto-dodge setting entails, before proceeding with the execution of the original task which must be started over from scratch. The time will be wasted if the dodge or parry fails. This means that the auto-dodge setting is extremely important.

With auto-dodge at its maximum setting, everything the character does is setting them up to be able to dodge, parry, or deflect any forthcoming attack. This goes so far as to have the player continuously stand and stand up in order to better be able to deal with attacks from a close melee combatant, by doing lunging dives among other things. If the attacks are coming for ranged weapons at longer ranges, the character will continuously kneel or go prone in order to present a smaller target. Both of these moves will give the player with the best possible saving throw to dodge or parry an attack, or the best cover modifier to avoiding being shot by a blaster shot.

With auto-dodge at its minimum setting it is turned off, and the character won't bat an eye as blaster-bolts scream past their heads or a light-saber's blade slices toward them. This means no parrying, and no dodging of incoming attacks.

At in-between auto-dodge settings, incoming attacks have to be more and more likely to connect before the player will try and dodge or avoid the blow and the player will be less and less inclined to sacrifice their current action unless the attack is more violent or meets certain criteria.

As an example, a player who is content to wait for their opening might have their auto-dodge setting maxed out until their opponent tires themselves or starts conserving their ammunition, then lower it significantly in order to take more shots.

How Cautiousness and Auto-Dodge Inter-relate

The two settings of cautiousness and auto-dodge are extremely powerful in controlling how a player fights. If both auto-dodge and cautiousness are maxed, the player's character is likely to avoid many blows, but they are probably so busy doing this that they are unable to throw many blows or make many shots.

If both settings are minimized, the player's character is ignoring the attacks the enemy makes upon their person, but their arms are a blur of action as they attack in a frenzy throwing attack after attack at the enemy, or jerking the trigger of their blaster as fast as they can.

If one setting is maxed and the other minimized, the player's character is either ducking every blow and punching as fast as possible, or they are carefully planting a powerful punch on the jaw of their enemy.

In terms of ranged combat, opposite cautiousness and auto-dodge settings could mean either a character leaping into cover while jerking their blaster's trigger as fast as possible in the direction of their enemy, or ignoring blaster fire hitting all around them while they coolly sight down the barrel of their blaster and draw a good bead on their opponent.

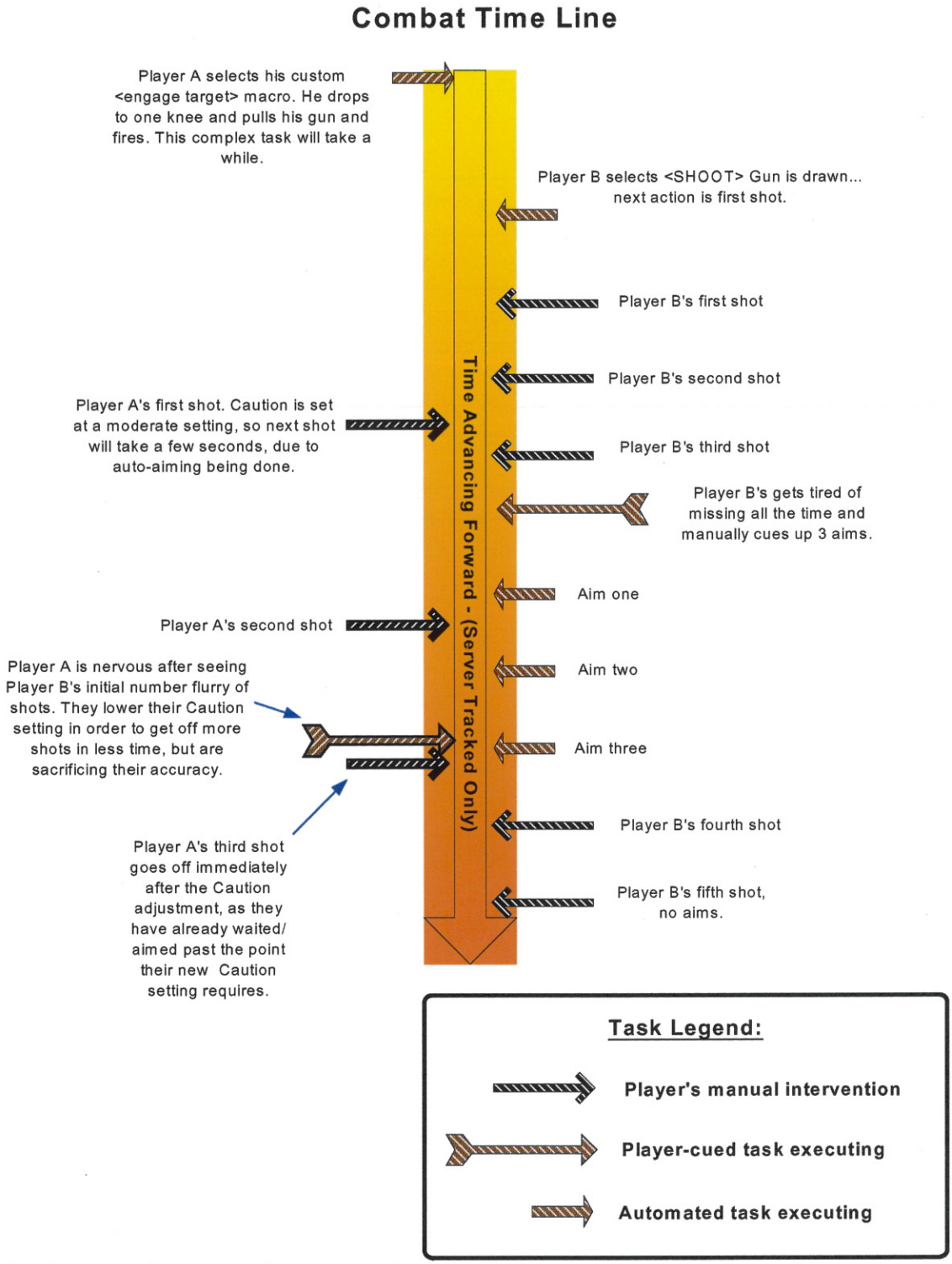
Advanced To-Hit Modifiers

While the basic combat skill-checks will base their difficulty primarily upon the character's skill, the range of their target, and the weapon being used, adding more conditions to check allow us to increase the depth of game-play when allowed to by server-latency situations. Examples of enhanced skill-check modifiers include movement and cover.

If a character is riding around on a speeder bike at full speed, it should be harder for them to hit their targets and be hit. Due to latency concerns, we wouldn't likely be able to assess modifiers based on movement, but rather on movement trends, like assessing a penalty based upon their average movement over the past segment of time.

If a character is standing in brush or behind a tree, they should be more difficult to hit than something standing in the middle of an open desert. The same situation occurs in built-up areas. If there are many small structures around, or the character is standing behind a vehicle or other construction, they should be harder to hit and we can

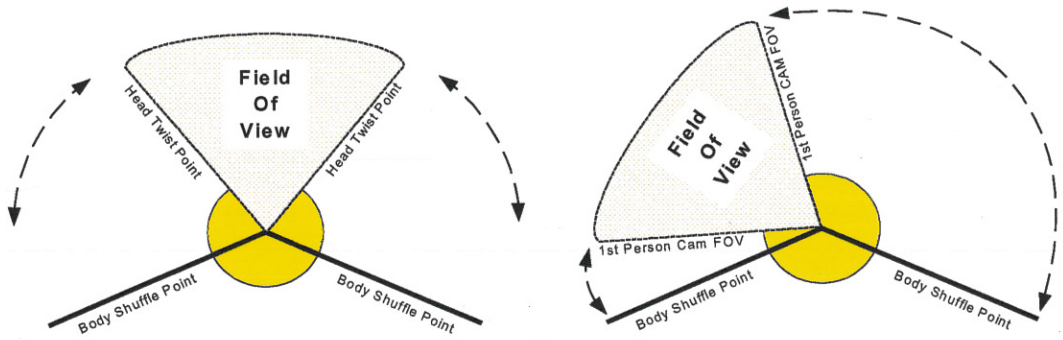
assess a skill check penalty for those situations. We would be approximating these penalties based upon the average terrain in the immediate vicinity, in order to avoid having to actually do collision-detection calculations.


Combat Time Line
Player A selects his custom <engage target> macro. He drops to one knee and pulls his gun and fires. This complex task will take a while.
Player B selects <SHOOT> Gun is drawn... next action is first shot.
Player B's first shot
Player B's second shot
Player A's first shot. Caution is set at a moderate setting, so next shot will take a few seconds, due to auto-aiming being done.
Player B's third shot
Player B's gets tired of missing all the time and manually cues up 3 aims.
Aim one
Player A's second shot
Aim two
Player A is nervous after seeing Player B's initial number flurry of shots. They lower their Caution setting in order to get off more shots in less time, but are sacrificing their accuracy.
Aim three
Player B's fourth shot
Player A's third shot goes off immediately after the Caution adjustment, as they have already waited/ aimed past the point their new Caution setting requires.
Player B's fifth shot, no aims.
Time Advancing Forward - (Server Tracked Only)
Task Legend:
Player's manual intervention
Player-cued task executing
Automated task executing

Combat Time-line: How the automated tasks and player-cued tasks work together to control the flow of combat.

Target Track

If so selected, the player doesn't have to manually keep his target within is camera's field of view. Every time the target strays out of his primary engagement field, the character will shuffle position to face it. Depending on how the character is situated and what equipment or weapon they are using this can take a second or two or no time at all. For example, even a strong character lying prone using a heavy-blaster on a bipod will take almost half a minute or so to turn to face a target while a character standing using a light-saber would be able to whip around almost instantaneously.



Interface

The combat displays and controls will be shown around the sides of the player window. They include all of the information displays and controls that the player requires to fight effectively, with the player being able to customize how much or how little information that they see. Some of the controls that they might have include:

- Ability to select their current degree of aggressiveness and defensiveness.
- List of moves they can cue up, and a list that shows what is currently cued up.
- Ability to arrange and save a set of moves. Should be able to arrange a 'favorites' palette of moves.
- Should have some readout of current task and estimated time till completion. Could be a palette changing bar.
- Need a weapon Icon to show current weapon state (aiming, loading, charging, getting set, whatever)
- Aim constraint indicator; are they facing their current target
- Selected target/targets list. Also, the ability to select and set up a prioritized list.

Revision Log

- Rev 1.1 – Changed doc to fit new template. Split details in to basic and advanced versions of crafting.
- Rev 1.2 – General Cleanup in pages.

Weapons List

Source Abbreviations

AESB	Art of Empire Strikes Back
BOLA	Best of LucasFilm Archives
CRFG	Cracken's Rebel Field Guide
ESWT	SW: Essential Guide to Weapons and Tech
SWRPG	West End's Star Wars RPG Book
SWTVD	SW: The Visual Dictionary
TRAS	The Rebel Alliance Sourcebook

Note: No number means no picture reference available. Mentioned only in text.

Categories Covered

Range Weapons	**Melee Weapons**	**Explosives**
Pistols	Batons & Clubs	Grenades & Thermal Detonators
Carbines	Staves, Lances, & Pole-arms	Mines
Rifles	Knives	
Light Support Weapons	Axes	
Shoulder-Launched Weapons	Swords	
Add-on Accessories	Lightsabers	
*Special Case Items	*Special Case Items	*Special Case Items

Additional Customization Thoughts

While specific alteration zones types are listed for the guns, it is possible that more specific and minor cosmetic alterations are available than those listed here. Possible examples include:

Custom grips
Metal Coloring
Metal Polishing

Ranged Weapons

PISTOLS	*Reference*	Scope	Barrel	Mag/Pack
Prax Arms Protector PRP-502 Hold-Out Shooter	ESWT 37			
Coruschal WeaponsWorks "Dagger" Combat Pistol		X	X	X
Oriolanis Striker Projectile Pistol	ESWT 33	X	X	X
Malaxan Firepower FWG-5 Flechette Pistol	ESWT 39	X		
Short Range Assassin's Blaster	SWTVD 7			
Tenloss DX-2 Disruptor Pistol	ESWT 25		X	X
Misc Unknown Pistols	ASW 70	X	X	X_E
Unknown Pistol	ASW 96	X	X	X_E
Pacnorval Defense Systems, Limited. SD-77 Sonic Pistol	ESWT 27			
Merr-Sonn Munitions Q2 Hold-Out Blaster				
Czerka 411	TRAS			
BlasTech DL-18 Blaster Pistol	TRAS 99			
BlasTech DH-17 Blaster Pistol	ESWT 5 & SWTVD 26 & BOLA 24	X	X	X
Scout Trooper Pistol	SWTVD 39	X	X	X
Skiff Guard's Blaster	SWTVD 47	X	X	X
Unknown Jawa Pistol	ASW 34	X	X	X_E
Pacnorval SIL-50 Sleep Inducer				X
Drearian Defense Conglomerate Defender	ESWT 11	X	X	X_E
Merr-Sonn Quick 6 Sporting Blaster	TRAS 99			
Short-range Combat Pistol	SWTVBD 34	X	X	X
BlasTech DL-44 Heavy Blaster Pistol	ESWT 9, SWTVD 14, TRAS 99, 100, BOLA 21	X	X	X
Merr-Sonn Flash 4	TRAS	X	X	X
Merr-Sonn Power 5	ESWT 21	X	X	X
Timms Emporer-4 Beam Tube	ESWT 21	X	X	X
Merr-Sonn DEMP Gun Pistol Type	ESWT		X	X
Merr-Sonn Anti-Riot Tangle Gun 7	CRFG 74			
Generic Pistol Body A		X	X	X
Generic Pistol Body B		X	X	X
Generic Pistol Body C		X	X	X

NOTE: X_E denotes extended length magazine/power pack is only modification possible.

CARBINES	*Reference*	Scope	Barrel	Mag/Pack	Bayonet	Stock	Bipod	Under-barrel Acces.
SoroSuub QuickSnap 36T	SWRPG 231							
Unknown Imperial Scout Blaster Carbine	BOLA 107							
Laser Carbine	SWTVD 7	X	X	X		X		
Merr-Sonn C-22 Flame Carbine	ESWT 35			X				
Merr-Sonn DEMP Gun Carbine Type	ESWT	X		X		X		
Tenloss DXR-6 Disruptor Rifle (actually a carbine)	ESWT 25		X	X_E		X		
Unknown Military Carbine	ASW 72	X	X	X		X		
Generic Carbine Body A			X	X		X		
Generic Carbine Body B			X	X		X		
Generic Carbine Body C			X	X		X		

RIFLES	*Reference*	**Scope**	**Barrel**	**Mag/Pack**	**Bayonet**	**Stock**	**Bipod**	**Under-barrel Acces.**
Laser Rifle	SWTVD 7	X	X	X	X	X	X	X
Pacnorval Defense System, Limited. Sg-82 Sonic Rifle	ESWT 26, 27							
SoroSuub Stormtrooper One Blaster Rifle		X	X	X	X	X		
BlasTech A280 Rifle	TRAS 99							
SoroSuub H.Tracker 16	TRAS							
Merr-Sonn G8	TRAS							
BlasTech E-11 Blaster Rifle	ESWT 7, SWTVD 34, BOLA 19	X	X	X		X		
BlasTech DLT-20A Laser Rifle	SWTVD 36	X	X	X	X	X	X	X
EE-6 Blaster Rifle	SWTVD 51	X	X	X		X		
Drearian Defense Conglomerate "Light Sport" Hunter		X	X	X	X	X		
Unknown Military Rifle	ASW 67	X	X	X	X	X	X	X
Unknown Civilian Rifle	ASW 71	X	X	X		X	X	
Jawa Ionization Blaster Type 1	ESWT 17, SWTVD 55, BOLA 19			X				
Jawa Ionization Blaster Type 2	ESWT 17			X				
Merr-Sonn CR-24 Flame Rifle	ESWT 35			X				
Generic Rifle Body A		X	X	X	X	X	X	X
Generic Rifle Body B		X	X	X	X	X	X	X
Generic Rifle Body C		X	X	X	X	X	X	X

LIGHT SUPPORT WEAPONS	*Reference*	Scope	Barrel	Mag/Pack	Bayonet	Stock	Bipod	Under-barrel Acces.
BlasTech T-21 Light Repeating Blaster	ESWT 13	X	X	X	X	X	X	
Generic Light Repeating Blaster A		X	X	X	X	X	X	X

PACK-ABLE SUPPORT WEAPONS	*Reference*	Scope	Barrel	Mag/Pack	Bayonet	Stock	Bipod	Under-barrel Acces.
BlasTech E-Web Repeating Blaster	ESWT 15, TRAS 101			X				
Merr-Sonn CR-28 Heavy Flamer				X				

SHOULDER-LAUNCHED WEAPONS	*Reference*	Scope	Barrel	Mag/Pack	Bayonet	Stock	Bipod	Under-barrel Acces.
Golan Arms FC1 Flechette Launcher	ESWT 39							
Merr-Sonn Munitions PLX-2M	ESWT 41							

**SPECIAL CASE ITEMS – RANGED*	*Reference*	Scope	Barrel	Mag/Pack	Bayonet	Stock	Bipod	Under-barrel Acces.
Prax Arms Stealth-2VX Palm Shooter	CRFG 72							
Ssi-Ruuvi Ion Paddle Beamer	ESWT 23							
Wookiee Bowcaster	ESWT 19	X						
Kelvarek Consolidated Arms MM9 Wrist Rocket System	ESWT 29							
Stokhli Spray Stick	ESWT 31							
Merr-Sonn MobileMortar-3	ESWT 47							

HEAVY WEAPONS / POSSIBLY INSTALLATIONS	Reference
Golan Arms DF.9 Anti-Infantry Batter	ESWT 71, TRAS
Loronar MAS-2xB	ESWT 49
Atrgar 1.4 FD P-Tower	ESWT 51
KDY v-150 Planet Defender	ESWT 99, AESB 38

WEAPON ADD-ON ACCESSORIES	**Reference**
Zone Control Viper Grenade Launcher	CRFG 72
Grenade Launcher Variant B	
Grenade Launcher Variant C	
Bayonet Variant A	
Bayonet Variant B	
Bayonet Variant C	
Barrel add-on: Muzzle Brake A	
Barrel add-on: Muzzle Brake B	
Barrel add-on: Muzzle Brake C	
Barrel add-on: Flash Hider A	
Barrel add-on: Flash Hider B	
Barrel add-on: Flash Hider C	
Alternate Power Pack A	
Alternate Power Pack B	
Alternate Power Pack C	

NOTE: We can share knives as bayonets. The key is that if a knife is to be used as a bayonet, its blade must be fairly straight. For example, scimitar would not be allowed due the blade curvature. A bayonet, when affixed to a range weapon, makes it useable as a spear or halberd. Additionally, the attachment point of the bayonet may be subdued or a large bracket.

NOTE: Alternate power packs vary in both size and shape.

Melee Weapons

BATONS & CLUBS	**Reference**
Merr-Sonn Stun Baton	ESWT 71
Fighting Club	SWTVD 56
Authority Stick	SWTVD 56
Gaderffii	ESWT 61, SWTVD 54, BOLA 20, ASW 34
Generic Club Variant A	
Generic Club Variant B	
Generic Club Variant C	

STAVES, LANCES, & POLE-ARMS	**Reference**
Vibro-Axe Pole-arm	SWTVD 23
Vibro-Lance	SWTVD 47
SoroSuub Controller FP	ESWT 63
Generic Staff	
Generic Pole-arm A	
Generic Pole-arm B	
Generic Pole-arm C	

KNIVES	**Reference**
Stone Knife	SWTVD 56
Survival Knife	SWTVD 51
Typical Knife	SWRPG 228
Typical Throwing Knife	
Twi'lek Dagger	SWTVD 46
Merr-Sonn Treppus-2 Vibro-Blade	ESWT 67
Generic Knife A	
Generic Knife B	
Generic Knife C	

AXES	**Reference**
Clan Groogrun Vibro-Ax	ESWT 65
SoroSuub BD-1 Cutter Vibro-Ax	ESWT 65
Heavy-Duty Ax	SWTVD 47
Generic Ax A	
Generic Ax B	
Generic Ax C	

SWORDS	**Reference**
Ryyk Blade	ESWT 69
Generic Sword A	
Generic Sword B	
Generic Sword C	

LIGHTSABERS	**Reference**
Darth Vader's Lightsaber	ESWT 57, SWTVD 6, BOLA 25
Luke Skywalker's Lightsaber	SWTVD 8
Luke Skywalker's Second Lightsaber	ESWT 57, SWTVD 6, BOLA 25
Obi-Wan Kenobi's Lightsaber	SWTVD 6
Generic Lightsaber A	
Generic Lightsaber B	
Generic Lightsaber C	
Generic Lightsaber D	
Generic Lightsaber E	
Generic Lightsaber F	

**SPECIAL CASE ITEMS – MELEE*	**Reference**
Czerka Vibro-Knuckle	ESWT 67
Exar Kun's Lightsaber	ESWT 59
Ryyk Kerarthoor Blade	ESWT 69
Aeramaxis Evasive-13 Individual Field Disruptor	ESWT 73
Aeramaxis Evasive-226-R Individual Field Disruptor	

Explosives

Grenades & Thermal Detonators	**Reference**
Standard Fragmentation Grenade	
Merr-Sonn C-22 Fragmentation Grenade	ESWT 45
Merr-Sonn C-16 Fragmentation Grenade	
Standard General Purpose Grenade	TRAS
Standard Anti-Vehicle Grenade	TRAS
Merr-Sonn Munitions 7-PrG Proton Grenade	ESWT 90
Merr-Sonn Glop Grenade	ESWT 45
Merr-Sonn WW-41 CryoBan Grenade	ESWT 45
Smoke Grenade	
Dye Grenade	
Luma Grenade	
Bothan Stun Spore Grenade	
Plank Gas Grenade	
T-238 Gas Grenade	
Fex-M3 Nerve Agent Gas Grenade	
Standard Issue Imperial Detonators	SWTVD 36
Merr-Sonn Munitions Class-A Thermal Detonator	ESWT 43, SWTVD 53, BOLA 93
Generic Grenade A	
Generic Grenade B	
Generic Grenade C	

MINES	**Reference**
Standard Anti-Vehicle Mine	
SoroSuub XG Anti-Gravity Field Bombs	CRFG 78
Nosam DR-X55 Repulsorlift Mine	CRFG 79
Merr-Sonn Defender Space-Based Ion Mines	CRFG 80
Generic Mine A	
Generic Mine B	
Generic Mine C	

**SPECIAL CASE ITEMS – EXPLOSIVES*	**Reference**
Micronite Explosive Charge and Timer Unit	CRFG 75
Blaster Power Pack Bomb	CRFG 76
Merr-Sonn Defender Space-Based Ion Mines	CRFG 80

GROUPING

Grouping offers players the opportunity for shared gain, mutual support, and purposeful communication. The Grouping System formalizes these features by providing players with sophisticated and friendly tools for group management and integrating them with core game systems.

Vision

In both space and on the ground it will be possible to form organized groups to share the benefits of kills, to make communication easier, and to promote community. In space these will take the forms of squadrons, with private comm channels and coordinated targeting systems. On the ground, it will be a more traditional RPG party system, including group tells and automatic splitting of loot and rewards. In addition, party members will always be visible on in-game radars and the like, via transponders, to make it easy to coordinate meeting up or attacking creatures.

Vision Checklist

- ❑ It should have an intuitive user interface. If the choice for initial interface is between flexibility and ease of use, we will choose ease of use, and make the flexibility an option.
- ❑ Players should feel empowered to take action against bad apples in the player base. They should never feel just like victims.
- ❑ Human administrative intervention must be minimized.
- ❑ We must have full support for localization, including Unicode support on client and server.
- ❑ New strategies should come to light for the player as their play time continues.
- ❑ *Star Wars Online* must provide multiple ladders of advancement so that new ways to play the game become apparent over time.
- ❑ Players should never feel ripped off by the game mechanics.
- ❑ We must encourage and support a community.
- ❑ Players need tools to shape their own experiences.
- ❑ Amy Jo Kim's principles for community building:
    - o We will design for a range of roles.
    - o We will facilitate member-run subgroups.
- ❑ Characters will be similar in power, but will gain more capabilities as they advance, so that players of disparate levels can fruitfully group together.

System Details

Forming a Group

Any player may form a group, and is automatically designated the group's Leader.

Naming a Group

A group begins with no (empty) name. The group Leader may name or rename the group at any time. Any news ascribed to a group uses the name and Leader name of the group at the time of the event.

Leader Abdication

The group Leader may abdicate their position by selecting a new Leader. The act of abdication will not disband the group, nor will it remove the former Leader from the group.

Group Benefits (or "Why be a good leader?")

A Leader with the proper skills automatically passes on bonuses to other characters in their group, and may include enhancements to combat and movement.

Leaders with high skill ranking will be able to lead larger groups of characters.

Leader Benefits

A group Leader is provided with additional interfaces to manage and view the status of the group members. This may include enhancements to the 'radar' interface.

Adding & Removing Group Members

Any group member can recruit new group members, subject to approval by the Leader. When another group member recruits a character, the Leader is provided with an interface to approve or decline the candidate. Recruitment events are queued, however only one outstanding recruitment event per group member is allowed.

The Leader may remove characters from their group at any time.

Players may remove their own character from a group at any time. If the Leader leaves a group, the group is automatically disbanded. The Leader should be reminded of this when they attempt to leave and offered the opportunity to abdicate and pass on the Leader position.

When recruited, a player is given the option to accept or decline. Any character may join a group. Joining another group automatically removes a character from any group they are in. **An exception to this may allow any Leader to join another group as a sub-group.**

Characters of opposing factions may join the same group, as there are no factional or reputation limitations for grouping. Outcasts may also create or join a group without restrictions.

Disbanding a Group

Only the Leader may disband a group. Subgroups, if we allow them, will automatically split and correctly 'reform' from a disbanded group.

Loot

Loot or physical rewards of any kind are not automatically distributed among the group members.

Achievements

Achievements are accrued individually, however a character can be assisted ('twinked') through a beneficial group association.

Faction

A group has no faction, per se; however, actions taken by any group member that affect faction standing will affect the entire group.

A group member initiating a factional event will be the most affected by the resulting faction adjustment.

Reputation

A group has no reputation, per se; however, the reputation (not UO-style reputation!) of individual group members may be affected by the actions of characters within the group. This will be decided on a case-by-case basis when the reputation system is designed.

Death & Looting

The death of any group member will not affect the status of a group. A Leader, even one spawning elsewhere in the game world, may abdicate his position to another living group member without disbanding or leaving the group.

Individuals have control over which other group members may loot their corpse, or choose to disallow looting altogether.

Member Status

- ❑ How is member status displayed to other group members?
  - o TBD

Features

- ❑ Radar display
- ❑ Member roster display
- ❑ Sub-grouping
- ❑ World Location of group members available to group.
- ❑ Persistent groups (maintained in friends list?)

Core Requirements

- ❑ Provide a system to form, lead, and disband a group.
- ❑ Provide a system to recruit and kick group members.
- ❑ Provide a system to communicate with individual group members.
- ❑ Provide any tie-ins to the Chat system, i.e. a group chat channel.
- ❑ Provide any tie-ins to the Combat system.
- ❑ Provide any tie-ins to the Faction system.
- ❑ Provide any tie-ins to the Reputation system.
- ❑ Provide any tie-ins to the Death system, including corpse looting.
- ❑ Provide any tie-ins to the interface for presenting member status.

Revision Log

1. 26-Sep-00 Draft revision (prior to design meeting)
2. 28-Sep-00 First pass

JEDI

The name Jedi, like the name Star Wars, is one that immediately invokes all kinds of feelings and expectations in fans everywhere. Our goal in Star Wars Online is to reproduce Jedi in such a way that people that wish to play a Jedi may strive to do so, while still preventing them from emerging as the dominant player-type and unbalancing the majority of game play.

Vision

The Jedi system is being designed to fit within the framework of the overall Star Wars Online experience, not to be the defining mechanism of it. SWO should roll along perfectly well if no one chooses to play a Jedi, nor should it screech to a halt if a great many would like to try their hand at it. In order to accomplish this goal, the designers continuously go back and revisit the original game Vision document, in order to be sure that our design fits what is intended for the final product. Listed below are some extracts from the SWO Vision document that are applicable to the Jedi system, as it is designed.

Issues for Newbies

- The game must be accessible to a broad audience.
- It should have an intuitive user interface. If the choice for initial interface is between flexibility and ease of use, we will choose ease of use, and make the flexibility an option.
- The game should offer a sense of discovery.
- The play environment should be customizable so that the players can change it to suit.

Live service issues

- We must have an average subscriber lifespan of twelve months or more.

Retention

- New strategies should come to light for the player as their play time continues.
- *Star Wars Online* must provide multiple ladders of advancement so that new ways to play the game become apparent over time.
- No one profession, specialty, or skill will be optimal—we will use "rock-paper-scissors" techniques to balance differing capabilities.
- Player-driven content will be a key element of the game.
- Experienced players will receive special privileges (such as playing Jedi).
- The game system will encourage specialization.
- We will explicitly reward non-advancement based ways of playing the game.

Gameplay issues

- The game must satisfy each of Bartle's Four: Achievers, Killers, Socializers, and Explorers.
- Character building and advancement must be a key mechanic.
- We will scatter in as many little touches that add depth as we can. Wherever possible, these touches will relate to hidden or interesting aspects of the *Star Wars* universe.

- The player must feel smart.
- The game should have secrets and lore to it.
- There must be a sense of exploration.
- Experience.
- There must be defining moments that are easily remembered.

Social interaction

- We must encourage and support a community.
- Players need tools to shape their own experiences.
- There should be ladders and leader boards for everything in the game, including social activities.
- Robust support for guilds and other such player organizations is a requirement.
- Amy Jo Kim's principles for community building:
  - We will create meaningful and evolving member profiles.
  - We will design for a range of roles.
  - We will develop a strong leadership program.
  - We will encourage appropriate etiquette.
  - We will integrate the rituals of community life.
  - We will facilitate member-run subgroups.
- Characters will be similar in power, but will gain more capabilities as they advance, so that players of disparate levels can fruitfully group together.
- The advancement model will be more character-driven than equipment driven so that changing equipment does not change your capabilities radically.

Terms

Jedi – (You're joking, right?)

System Details

Getting Jedi into the game without knocking everything out of whack will be a bit of a challenge. There are solutions to every problem. Some better, some worse. The ones presented herein seem to be the best compromises.

Core Requirements

The Jedi system requires more stuff to get it into the game than some others. This is fitting, considering how critical they are the genre, and how destabilizing they could be to game play in general if not properly implemented. The following sub-sections are a brief run-down of what we'll need to do to get Jedi into SWO.

Selection of Jedi

Becoming a Jedi should 'feel' special
It requires many and varying criteria to be fulfilled
Jedi status can be lost

Mysterious and rare. These are two of the traits that we want Jedi to have. We don't ever want Jedi to make up the dominant number of Star Wars Online's in-game population and we don't want those that have become Jedi to be entirely sure of how they attained that status. At the same time, we don't want to lock anyone out of the

system that wants to strive to achieve the role and status of Jedi. Toward this end, there is no artificial cap to the number of Jedi that can be present in the game at one time, but rather a long and dynamic list of criteria that must be satisfied before they can even be considered for the role, in addition to mechanisms that will act to control the actions and reduce the numbers of Jedi present.

So, you want to be a Jedi

A Jedi isn't an archetype that can simply be selected when a player creates their character. That would be too easy. If we did that, we'd have lightning-bolt-throwing grief players looking for a single evening's satisfaction running around all over the place telekinetically dropping boulders on people's heads. Although any character can evolve into a Jedi over their lifetime, it will not happen overnight and the experience should vary from player to player. This serves to make being a Jedi more a bit more special.

As mentioned before, the number of Jedi that can exist at one time is effectively unlimited. We aren't planning on putting blanket maximum of '5%' of the player population can be Jedi, or anything of that nature. The main reason for this is that we don't want people that join SWO after it's been rolling along for several months to be locked out from being Jedi because all the slots are used up. The Jedi population should wax and wane. Due to the dangers inherent in being a Jedi and maintaining one's Jedi status, it will be a struggle to remain a Jedi, and players will not be able to simply rest on their laurels if they wish to keep the status. Still, we want anyone that's willing to attempt to fulfill the criteria we have set forth to have a chance at being one. It won't be easy.

The first and most basic criterion is longevity. A character will have to have been active for least a certain period of time before they can be considered for being a Jedi. The current figure we are looking at is three months of real time (not game time). This should help trend toward players that are in the game for a long haul instead of an insta-thrill. The time delay adds a bit of a catchall shock-absorber mechanic that helps act as a bit of a safety net for the rest of the Jedi growth system.

After fulfilling the longevity requirement, a character is checked against a portion of a long list of metrics and criteria that are being tracked for each character in order to see if they measure up to what we have set as the 'Jedi' standard. The purpose of the entire metric list can be summarized as attempting to verify that a character is 'well-rounded'. This includes such things as having to have played the role of each of the four Bartle's player types: Achievers, Killers, Socialites, and Explorers.

The criteria list itself will include around two hundred or more metrics. These include such things as:

player killing, thievery, mining, use of moods, achieving a maximum in a skill category, use of the buddy list, use of the chat system, been a group leader, helped a newbie, had a high reputation in something, joined an association before, completed a mission before, and more. Out of this master list of two hundred criteria, a character will be designated to track a randomly chosen one hundred of them. This will aid in avoiding situations where the process of becoming a Jedi can be summarized accurately as a step-by-step process on a website somewhere. What's more, two characters should be able to talk about how they became a Jedi, and be unable to pin down what brought it about. Although we will initialize the system, the Beta cycle will play a valuable role in helping us to fine-tune the criteria in order to get exactly the kind of performance we want.

Once a Jedi, always a Jedi?

Not necessarily. Just because you've finally attained you're Jedi-dom, doesn't mean you can run around bragging, shooting off fireworks, and taking on entire legions of Imperials or Rebels, secure in the knowledge that now that you've attained your lofty goal you can lord it over everyone without fear of losing it. Neither does it mean you can quietly retire to Florida and be a Jedi beachcomber. Nope.

Losing a character's Jedi abilities is easier to do than it is to attain them in the first place. This doesn't mean that maintaining Jedi standing is incredibly difficult to do or particularly time consuming. It just means that maintenance must be done, and that a character playing the role of a Jedi cannot indiscriminately call upon the full might of their powers.

There are several types of maintenance that are necessary to be a Jedi. Some maintenance is tied up in training, much like it is for maintaining a particular skill. Some maintenance means fulfilling obligations your current level and classification of Jedi have to your faction or community. Finally, some maintenance is done by limiting your actions to ones that are 'in character' for your role such as not belittling your powers and becoming a Jedi janitor. A Jedi will be required to play an active role in conflict, either working to settle disputes or working toward starting them. It won't be necessary to fill your 'light-saber duel' quota necessarily, but training fellow Jedi is about as far from the conflict as a Jedi can be.

If you die, your clone (if you have any remaining) comes to life, sans the Force. To say it again, whenever a character is killed, their force-sensitivity stats and metrics are reset. This does not include the minimum character life requirement (three months, currently). There is nothing to prevent a Jedi that was waxed and lost their power from regaining it; it's simply going to take a bit however long it takes them to get their custom criteria listing in shape.

Just because you're paranoid, doesn't mean they aren't out to get you

The Empire is out to get Jedi's. They want them all to be ex-Jedi's. Semi-covert teams of assassins and hit squads are out and around space, looking for Jedi's. As a result of this, Jedi's that make spectacles of themselves in the wrong areas will need to move along in short order.

Whenever a Jedi uses an 'active' force power in front of characters or NPCs of a differing faction, those characters will have the option to 'report' the Jedi to their faction, and gain a minor faction and/or reputation bonus. An active power would be one that has a fairly visible manifestation of itself, such as levitating objects or using a lightsaber

with Zorro-like ability. When this happens in visual proximity to any characters of dislike factional standing, those characters have an option window pop up that gives them the ability to 'report' the Jedi. These reports accumulate, and the more there are on a Jedi, the more hurt is called down on them.

A character that only has a smallish number of reports filed against them, between one and twenty-five perhaps, is perceived as a minor nuisance and has both NPC and PC Bounty Hunter missions filed against them. The amount of the bounty and number of hunters sent would vary with the target character's reputation and number of reports.

A character that has a mid-number of reports filed against them, between twenty-six and fifty perhaps, would have more elite teams sent against them, in addition to the already present bounties. These elite teams would include some of the more famous (or infamous) bounty hunters of the Star Wars universe, and would also include packages of troopers as well as other more highly skilled assault teams.

A character that has a large number of reports filed against them, around fifty-one or more, would have serious hurt called down upon them. This could include having Vader or Mara Jade arrive to kick their butts, or any other large-scale nightmare we can think to sic upon an over-active show-off Jedi. Again, once a Jedi is fragged, they lose their power. If this happens, then the Jedi is no longer a threat, and their contracts are either cancelled or cashed out.

Advancement for Jedi

Once a player has developed the capability to deal with the force, they don't immediately become a Jedi and then set off to either wipe out tyranny or sow it. There's just a bit more to it than that.

The answer to Life, the Universe, and Everything? Five, of course.

There are five stages in a Jedi's advancement. Unlike the character age requirement and secret criteria it takes to become force-sensitive in the place, the steps to follow and stages to advance through Jedi-dom and beyond are made fairly clear to characters that have reached a minimum capability of force-sensitivity.

- Force-Sensitive
- Padawan
- Jedi Knight
- Jedi Master
- Jedi Council

Force-Sensitive

A character that meets the minimum game-lifetime requirement and fulfills the requirements of the one hundred or so secret criteria that were randomly assigned to them instantly becomes Force-Sensitive. Force-Sensitive is the first stage of becoming a user of the Force. Force sensitivity will show itself in the player beginning to see manifestations of the Force and its use on their game screen. This will varying from seeing auras of varying strength and dark or lightness around other force-sensitive or using characters, to seeing flashes or waves of it emanating from locations where it used to varying degrees. If a force-sensitive player is in proximity to a force-sensitive light-sider, then a faint light aura will be visible around that character. If a force-sensitive player were in proximity to Vader, then an extremely prominent, large, and dark aura

would be visible around him. If a character witnesses a character levitating an object, they would see auras around the character and object, as well as lines of force between the two. The prominence of the auras would depend on how heavy the object was. If the Emperor were to use his might to fry another character nearby but off-screen to the Force-sensitive player, then the player would notice a flash and aura coming emanating from the direction of the use of the Force. On top of these abilities, characters can also tell if something is alive or dead by checking out its aura, and cataclysmic events also grab their attention. A character is less likely to notice these effects if they are fatigued or currently concentrating on a task, although the detonation of Alderaan would briefly light up the sky like a lightning bolt.

Padawan

To advance, a Jedi needs either a Jedi Knight or Jedi Master to apprentice them. This could involve the Force-sensitive character looking for someone to teach them, or a Knight or Master asking them to be their student. A Padawan is the proper name for an apprenticed Jedi, and is a stage during which they must learn first-tier Jedi powers. In order for a character to advance to the Padawan stage, the character accepts an offer to apprentice from either a Jedi Knight or Jedi Master, gives up claim to all the plots of land a character are initially given, and can only ever possess or lay claim to a subsistence level of money. After a Padawan has learned a minimum allotment of first-tier Jedi powers, they can advance to being a Jedi Knight by completing a series of quests that culminate in their building of their own, personal lightsaber. At this point it is up to the membership of the current Jedi Council to vote on the character's acceptance into the ranks of the Jedi, or to continue their training.

Jedi Knight

The majority of Jedi are Knights, having completed their Padawan training cycle and been granted their title by the Jedi Council. At this point, they must begin to fulfill the minimum mission obligations as determined by the Council or by the SWO game systems. Jedi Knights are also eligible for learning second-tier force powers. The Jedi Knights are the rank and file of the Jedi.

Jedi Master

Jedi Masters are elder Jedi that have been rated as full Jedi Knights for a minimum period of time, have trained a minimum number of Jedi Knights, and are voted into the position by the Jedi Council. Jedi Masters are the elite of the active Jedi cadre and have both training as one of their primary obligations and directives. Jedi Masters are eligible to learn the third-tier Force powers. Jedi Council members are elected from the body formed by the Jedi Masters.

Jedi Council

The Jedi Council is the controlling body for the Jedi. They are a politically run organization that is allied and is often subordinate to the governing body of the New Alliance. Although a great many immutable traditions and rituals are already laid down and followed exactly by the Jedi, the Council can still generate special missions for the rank and file Jedi to complete, as well as having the final responsibility of conferring the status of Jedi Knight and Jedi Masters onto candidates. The council has regular elections in which members are nominated from and elected by the body of the Jedi Masters. Thus, the council is made up of player characters. In order to be a Jedi Councilperson.

Jedi Powers

There are a great many of Force powers with varying in-game effects. They are attained through many different means. The most basic and least powerful are first tier powers can be learned by a Padawan from their teacher, regardless of that teacher's personal knowledge or capabilities. More powerful first tier Force powers are only known by some Jedi and can only be taught by those Jedi with the knowledge. Once a character has become a Jedi, second tier powers become available to learn. A Jedi Master as access to the most powerful powers, which are known as third tier powers. Each successive tier of powers is a combination of enhancements to lower-level powers, and some that are entirely new.

A force power itself is treated as a skill tree that is available only to users of the Force who have attained a high enough level of power and control to be able to utilize it. Some powers a merely available or not, while others can be improved to higher levels of capability after having been learned at their base power level.

Some powers include:

- ❑ Telekinesis
- ❑ Telepathy with Jedi or force sensitive being
- ❑ Improve accuracy
- ❑ Deflect bolt with hand
- ❑ Block bolt/redirect bolt with lightsaber
- ❑ Absorb force attack
- ❑ Force strength
- ❑ Force lightning
- ❑ Force jump
- ❑ Force speed
- ❑ Force flip
- ❑ Force push
- ❑ Force pull
- ❑ Force throw
- ❑ Force choke/grip
- ❑ Force destruction
- ❑ Enhance reflexes
- ❑ Battle meditation
- ❑ Heal wounds
- ❑ Cleanse poison
- ❑ Hold breath
- ❑ Resist fatigue
- ❑ Hibernate
- ❑ View potential past/alternate past
- ❑ Foresee future
- ❑ ESP
- ❑ Mind alter
- ❑ Sense feelings
- ❑ Group influence
- ❑ Area awareness/hyperconscious
- ❑ Force seeing
- ❑ Force Blinding
- ❑ Enhance memory accuracy

- Force sensitivity/awareness
- Become luminous being (blue glowy dude)
- Speak with beasts

The Dark Side

There are times where a character's actions run contrary to the nature of 'lightside' Jedi. Normal Jedi are out to stamp out Evil in all of its forms, and avoid doing evil deeds themselves. For example, a Jedi can never kill, except in the case of immediate danger of loss of life.

Dark side-ness feeds upon itself. The use of 'Dark-sidey' powers makes a character darker.

One type of dark side powers is a perverted form of regular powers that have been 'misused'. The other form of dark power is one taught by a dark Jedi Knight or Master.

Dark Jedi have their own controlling body, known as the Sith. There are only two Sith and they are both NPC controlled characters. A player character can never become a Sith Lord. Instead, all dark Jedi are members of the Brotherhood of Sith, who are themselves responsible to the Sith Lords.

Interface

The Jedi system pulls from pre-existing interfaces in all areas. Council control is exercised using the political interface screens used by other player-governed bodies, and Jedi power selection is done using the screen used for selection of skills.

Assets Required

Art for Jedi Power effects.
Art for various Jedi Power icons.
Theme Art for Jedi council interface (borrowed from other political interfaces).

Revision Log

Rev 1.0 – Document Generated.
Rev 1.1 – Fleshed out intro to 'Vision' section.

Questions remaining:

Can you beat Vader or Boba Fett if they are called after you?
Can you lie low and have people forget the reports they called against you?
Does the Alliance hunt down Dark Jedi like the Empire does?
How does a Jedi maintain their Jedi-ness?
If not, then why doesn't everyone become dark Jedi? It's an easier life.

Does the dark side of the Force damage your body? Cost more force points to use powers? Cause damage?

Does the implantation of cybernetics lower a person's force power?

Council votes are a popularity contest, aren't they?

STAT ADVANCEMENT

The goal of the statistics advancement model in SWO is to provide a balanced statistics system, while offering the player an opportunity to customize those statistics – over time – to coincide with their vision and goals for their characters.

Vision

Key design goals are to allow players to change professions at any time, and to make players have to work to keep up their standing. Additional goals are that the power differential between two characters in terms of raw statistics and damage done not be great, but that advancement be primarily via acquiring more capabilities, and that equipment not be the primary driver of advancement (e.g., a character who is naked does not lose most of her capability)..

- The game must be accessible to a broad audience.
- It should have an intuitive user interface. If the choice for initial interface is between flexibility and ease of use, we will choose ease of use, and make the flexibility an option.
- We must impress the user within 30 seconds via dramatic visuals, impressive use of sound and music, and compelling interaction.
- The new player must feel rewarded and smart within the first five minutes.
- A high degree of customization for both players and ships is a key requirement.
- New strategies should come to light for the player as their play time continues.
- No one profession, specialty, or skill will be optimal—we will use "rock-paper-scissors" techniques to balance differing capabilities.
- There will be motives and rewards for short session play.
- Character building and advancement must be a key mechanic.
- Players should never feel ripped off by the game mechanics.
- Players must frequently feel rewarded.

System Details

There is a base stat advancement rate for all players. The advancement rate of a statistic towards a goal is affected by a 'burst hour' of play per day which will directly benefit the casual gamer, balancing the rate at which hardcore gamers and casual gamers progress in statistics. Stats account for approximately 20% of success or failure in all skill system checks.

Requirements

Initial Statistics

All characters start with statistics appropriate for their species. The value of the combined statistics will leave enough points to equally enhance two other attributes. Once enhanced, the player will have received ALL statistic points available to them.

Character Creation

During character creation, the player must be provided with an opportunity to choose two enhanced attributes. The value added to these attribute is equal to one-half the remaining points. The players shouldn't have to think of these in terms of points, but as simple stat selections.

Statistics Advancement

A character begins with all the statistic points ever available to them. Statistics Advancement occurs through realigning the points from one statistic to another. At any point during the game, a player may realign his stats. This process sets one stat goal to be achieved over a fixed period of time. The time to reach a stat change of P points is the same for every character.

Features

Certain achievements during the game may allow the player to reach a stat goal more quickly. Once achieved, the player may not attempt a similar achievement again until a length of time has passed equal to that of reaching the original stat goal. Modifying the stat goal after the fact does not change this constraint.

Interface

Although not as relevant to this overview, interface screens for character creation and stat modification will be required.

The interface must provide a way to set one primary statistic goal, as well as locking a statistic or lowering a statistics goal to advance towards a primary statistic goal.

Change Log

1. 16-Aug-00 Initial revision.
2. 28-Aug-00 Added System Details introductory paragraph.
   Added more conceptual interface requirements.

SKILL ADVANCEMENT

Welcome to the SWO Skill System Overview. This section of the document is meant to give any reader a baseline understanding of how the skill system will work. It is not intended to provide detailed information on what skills are available and how each one is used.

Overview

SWO's skill advancement system is classless. This means that no matter what species or other character attributes a player decides upon during creation, the character has access to all skills trees.

Skills are arranged in trees based on association. For example, all the skills of the culinary arts will be arranged together. You won't be able to upgrade your "bantha stewing" skill to "blaster level 4".

Players can obtain titles based on their mastery of high level skills. Players will always keep these titles. If they ever give up the skill, their title will have "former" in it. For example: Jun Qwa, "Master of the Culinary Arts" may become Jun Qwa, "Former Master of the Culinary Arts", if the player decides to give up some of their culinary skill for droid repair. The purpose of tracking former titles to allow players to change their style of gameplay while still keeping a record of their previous conquest. This will also allow us to let the player move up the lower levels of alternate skill trees quickly.

Vision

- New strategies should come to light for the player as their play time continues.
- *Star Wars Online* must provide multiple ladders of advancement so that new ways to play the game become apparent over time.
- Players should be able to switch professions or advancement ladders without having to start over.
- No one profession, specialty, or skill will be optimal—we will use "rock-paper-scissors" techniques to balance differing capabilities.
- There will be motives and rewards for short session play.
- Experienced players will receive special privileges (such as playing Jedi).
- The game system will encourage specialization.
- Players should have to work to remain at their current level of achievement.
- We will explicitly reward non-advancement based ways of playing the game.
- The game must satisfy each of Bartle's Four: Achievers, Killers, Socializers, and Explorers.
- Character building and advancement must be a key mechanic.
- Players should never feel ripped off by the game mechanics.
- There must be a sense of exploration.
- Characters will be similar in power, but will gain more capabilities as they advance, so that players of disparate levels can fruitfully group together.
- The advancement model will be more character-driven than equipment driven so that changing equipment does not change your capabilities radically.

Terms

Skill Box: Refers to a specific skill.

Profession: A profession is a designer term that assists developers in fleshing out the skill trees. Profession names usually imply fairly concrete skills sets. They are therefore useful in creating lists of skills and outlining skill relationships.

Title: Some skills confer a tile upon the player. Usually these skills have a variety of skill requirements.

Skill Tree: A hierarchical list of related skills (also called categories or groups)

System Details

Skill Trees

The following excel spreadsheet provides an example of how "professions" can be used to create skill groupings. Originally, there was an "equipment" skill tree as well as a "vehicle" one. However, once we had assigned all the professions that had been brainstormed, we found that equipment had none. As a result, I have incorporated the vehicle tree into the equipment one. "Equipment" indicates the use of any tool or machine. Vehicles are subsets of equipment. This leaves room for expansion if we add more "equipment" professions or skills.

Skill Trees	Professions										
Athletics	Athlete										
Crafting	Cook Hairdresser Tailor	Cartographer	Armorer	Mechanic	Droid Repair	Chef Architect	Weaponsmith	Droid Maker Prostheticist	Shipwright	Engineer Inventor
Combat	Gunner	Explorer	Hunter Ranger	Soldier	SAR Detective Demolitions	Bounty Hunter Bodyguard	Squad Leader Spy Diplomat	Captain	Assassin	
Social	Writer	Dancer, Musician	Diplomat	Politician						
Science	Animal Tamer	Medic	Chemist Biologist Geologist	Biochemist Hacker	Doctor	Forensic Scientist	Xenobiologist	Geneticist		
Commerce	Merchant	Salvager	Harvester Bartender	Miner	Farmer	Rancher	Cloner			
Equipment	Pilot Driver	Space Pilot Air Pilot Naval Pilot	Mariner							
Languages	Linguist	Translator								

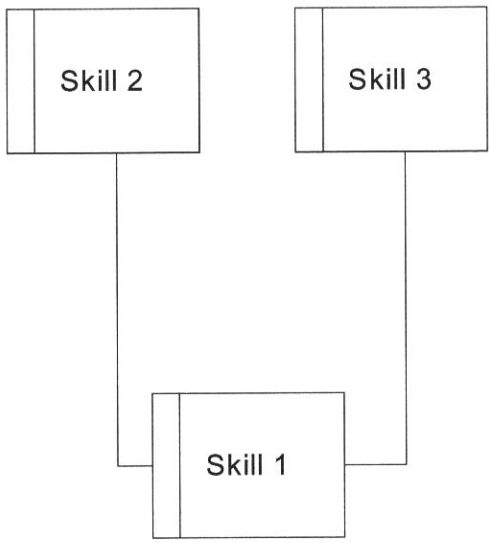
Skill Advancement

Players obtain new skills by meeting requirements in the following categories:

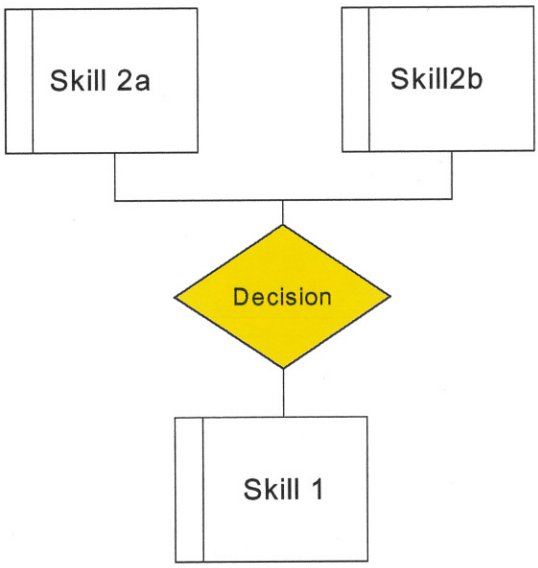
1. Purchase price: To become certified in a skill, the player must pay a certification fee. Generally, the higher level the skill, the more money it costs to obtain the skill.

2. Skill prerequisites: New skills may require that the player become certified in other skills first. These skills may be directly below the new skill in the tree, but in some cases (hybrid skills) the prerequisite skill may actually reside in an entirely different tree!
3. Usage requirements: New skills may require a minimum amount of recent usage in at least one of the prerequisite skills (see skill tree diagram).
4. Race: Some skills will require the player to be of a specific race in order to obtain it.
5. Mission: Some skills may require the player to complete a task before obtaining the skill.
6. Faction Standing: Some Certifications are only available from a particular faction (Imperial or Rebel).

Independent Skill Advancement:



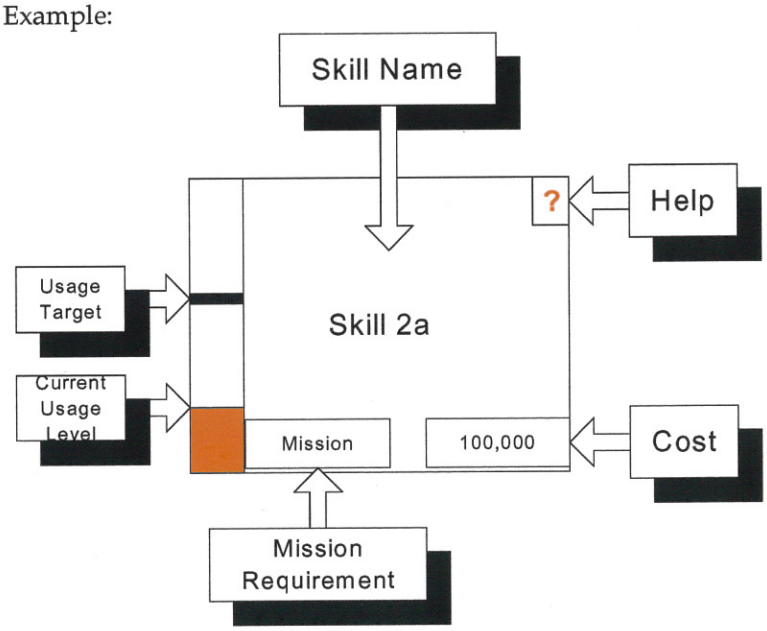
Branching Skill Advancement:



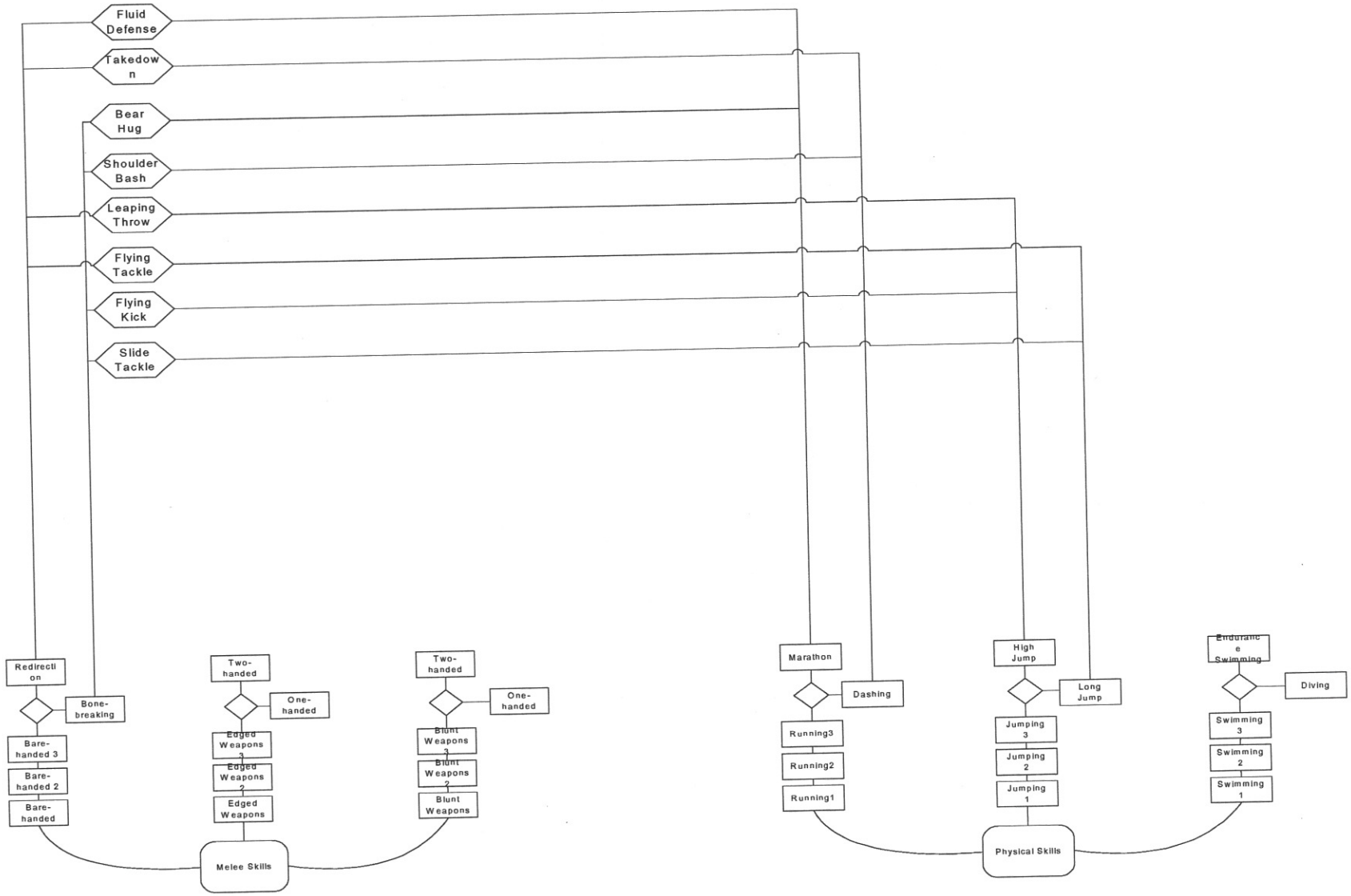
Interface

The players should be able to browse the entire set of skill trees. Trees should clearly illustrate the skill advancement path including what skills are required, the usage, and the purchase price. The interface should allow the player to pull up specific information about every skill.

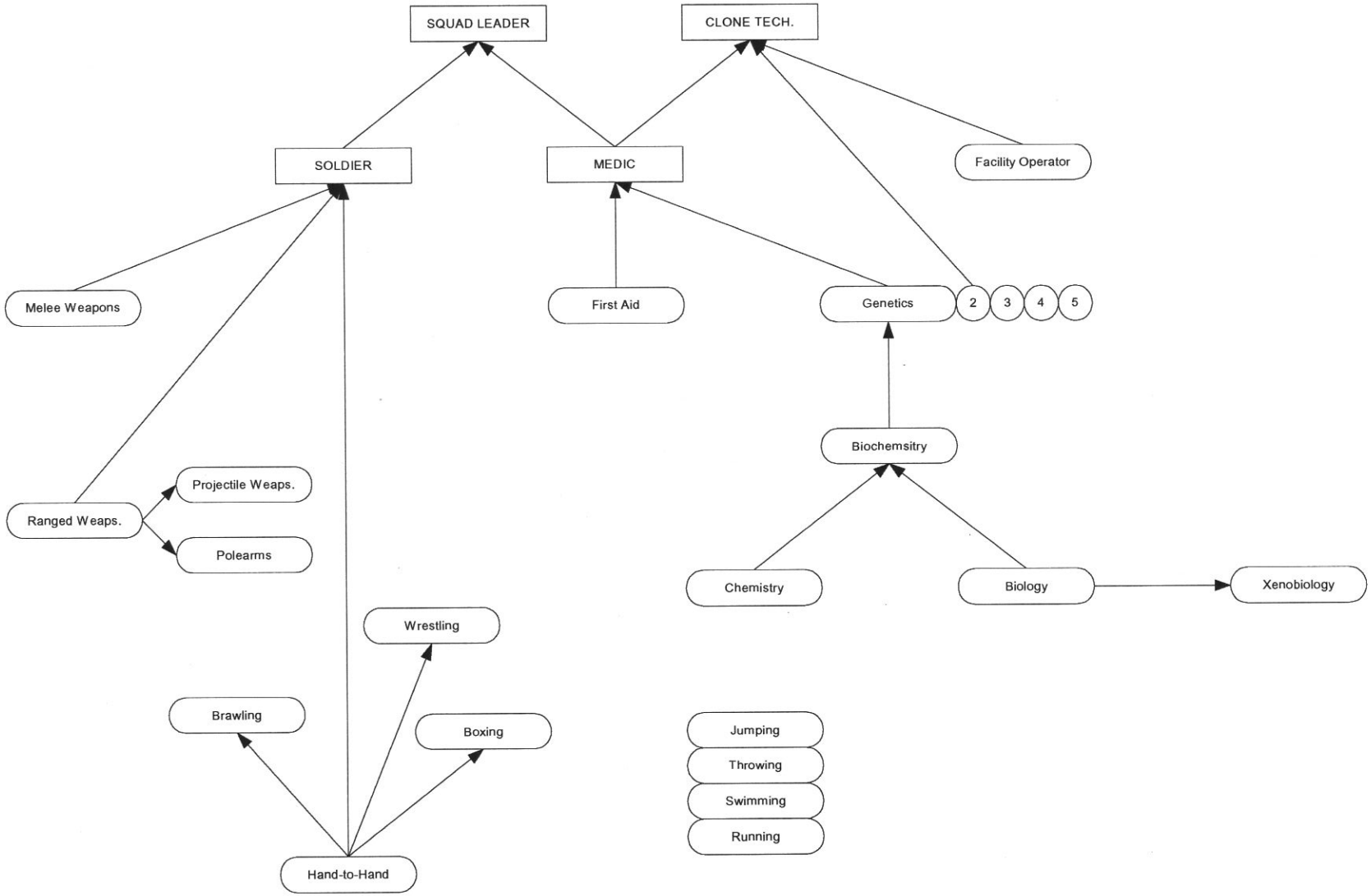
Example:



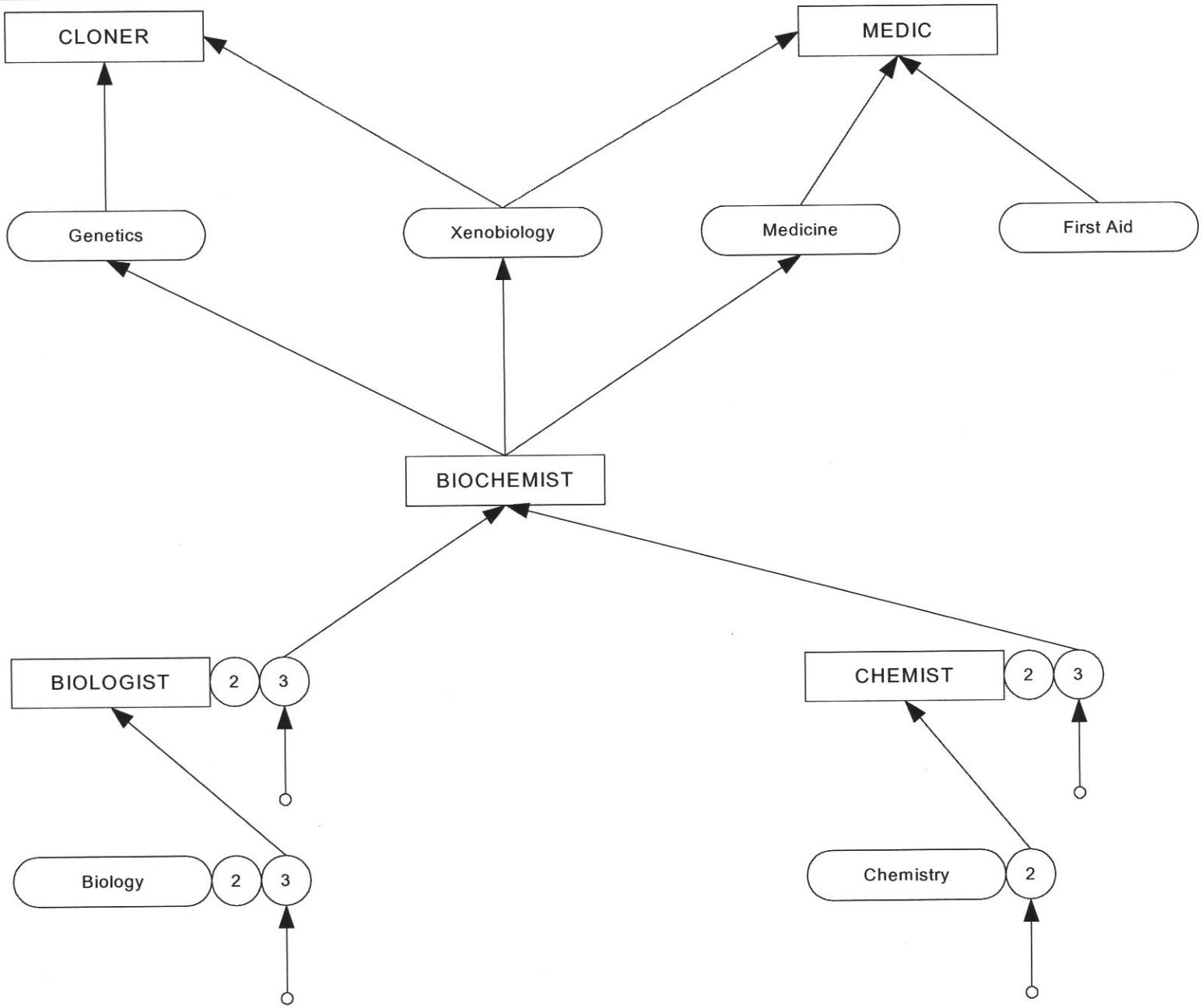
SAMPLE SKILL TREES



V t Confidential


SQUAD LEADER
CLONE TECH.
SOLDIER
MEDIC
Facility Operator
Melee Weapons
First Aid
Genetics
2
3
4
5
Biochemsitry
Projectile Weaps.
Ranged Weaps.
Polearms
Chemistry
Biology
Xenobiology
Wrestling
Brawling
Boxing
Jumping
Throwing
Swimming
Running
Hand-to-Hand

Tactics
Personal Defense
Dueling
Sharpshooter
Quickdraw
Melee
Hand-to-Hand
Brawling
Boxing
Wrestling
Melee Weapons
Dueling
Ranged Weapons
Staves
Polearms
Spears
Blunt Weapons
Clubs
Bladed Weapons
Knives
Swords
Projectile Weapons
Launcher Weapons
Thrown Weapons
Throwing Knives
Grenades
Energy Weapons
Non-lethal Weapons
Demolition
Explosives


CLONER
MEDIC
Genetics
Xenobiology
Medicine
First Aid
BIOCHEMIST
BIOLOGIST
2
3
CHEMIST
2
3
Biology
2
3
Chemistry
2

Missions System

The Mission System provides a broad array of Mission-related services, including mission browsing, player mission generation, quest creation, targeted mission filtering, instant contracts between players, mission rewards and penalties, and achievement tracking.

Vision

For both ground and space we will both allow players to enter missions into the database themselves (for example, to engage in commerce or espionage with their faction) or to take missions generated by the game itself. Most of the AI generated mission types will be fairly simple: retrieve item, kill target, destroy item, carry message, etc. The player-generated system will allow players to enter similar types of missions using a fill-in-the-blank system. The server will automatically track completion in order to award faction points or money. This allows players to get others to do Kessel runs for each other, or to spy on Imperial bases, and otherwise add content to the game for their own game playing purposes.

The Mission System ties directly in to many other game systems including Factions, Guilds, Grouping, Advancement, Death, Jedi, the Commodities Market, Banking, Trading, Cities, and evolving Character Profiles.

Vision Checklist

- ❑ The game must be accessible to a broad audience.
- ❑ It should have an intuitive user interface. If the choice for initial interface is between flexibility and ease of use, we will choose ease of use, and make the flexibility an option.
- ❑ The first thirty minutes of game play should surprise but always pleasantly.
- ❑ The game should offer a sense of discovery.
- ❑ It should be possible to play a satisfying game session in less than an hour.
- ❑ Players should feel empowered to take action against bad apples in the player base. They should never feel just like victims.
- ❑ We must have full support for localization, including Unicode support on client and server.
- ❑ It must be easy to extend and add content to the game after ship.
- ❑ New strategies should come to light for the player as their play time continues.
- ❑ Player-driven content will be a key element of the game.
- ❑ There will be motives and rewards for short session play.
- ❑ We must encourage and support a community.
- ❑ Players need tools to shape their own experiences.
- ❑ Robust support for guilds and other such player organizations is a requirement.
- ❑ Amy Jo Kim's principles for community building:
    - o We will create meaningful and evolving member profiles.
    - o We will design for a range of roles.
    - o We will facilitate member-run subgroups.

Terms

Achievement

An Achievement is the result of player action or interaction, the success of which can be determined by the game mechanics. Achievements are binary, either on or off, and therefore not measured on a scale. Instead, Certifications are used to measure the level of success. Achievements may result from any player action and are, therefore, ancillary to the Mission System.

Objective

An Objective is a tangible goal within the context of a *Mission* that can be accomplished within the scope of game mechanics. An Objective usually consists of a goal and a reward. In many ways, the goal can be thought of as a verb-noun combination with optional preposition. Kill Player. Deliver Item to Player.

Reward

A Reward is a tangible or intangible object supplied by the Mission Creator, and granted to the character accepting and completing a Mission to a level of success specified by the Objective.

Mission

A Mission is a collection of Objectives, generated by the game or created by a player, which may be offered to, accepted, and completed by other players. If a Mission contains several Objectives, one may be chosen to represent the context of the entire Mission.

Mission Package

A Mission Package is a collection of Missions associated by branching and/or linking as defined by a Mission or Objectives therein. A Mission Package may be submitted to the Mission System and only the first Mission in the series, as specified by the creator, will be posted to the Mission Board. Other Missions from the Mission Package are posted as Objectives or Missions are either completed or failed.

System Details

Mission Complexity

The simplest mission consists of a Mission Creator and an Objective. A simple Objective would be: Deliver an Item to a Person. A Reward need not be implicit, however most players will want something for their effort.

Missions with more complexity require additional data, and may be edited through advanced interface options.

Becoming Mission Leader

By default, player-created Missions can be accepted by anyone. The player/character accepting a Mission is considered the Mission Leader; this status does not and cannot change during the course of a Mission.

Mission Acceptance Filters

Mission Creators may choose to set filters on who may accept their Mission. These filters include settings for affiliation types such as Faction, Guild, and Town affiliation, as well as player-run Associations. A specific affiliation (such as a particular Guild) may also be used. Additional filters may require potential Mission Leaders to have specific Skill levels or Achievements to accept the Mission. This is an advanced feature.

Mission Exclusivity

By default, a Mission may be accepted by only one character before it is hidden from the browser, and subsequent attempts to accept the Mission will fail with an appropriate message. A Mission may be defined as non-exclusive, allowing multiple characters to accept and attempt the Mission. This is an advanced feature.

(Q: Can Missions be set up to be COMPLETED by multiple players? In other words, pay five different characters who accept and complete the same non-exclusive mission?)

Mission Expiration & Recurrence

Missions are one-time events by default. Once a Mission has been completed, successful or not, the Mission hooks are deleted from the Mission System, database, and any associated game objects.

Mission Creators may provide an expiration time for a Mission, and even create recurring Missions with a variety of scheduling rules. This is an advanced feature.

Mission Objectives

Each Mission has an Objective List that can handle multiple Objectives. Once the Primary Objective has been edited, Mission creators may insert or add other Objectives. This system has advanced options.

Critical vs. Non-Critical

By default, all Objectives are critical and must be accomplished to successfully complete the Mission. Any critical Objective may be flagged as the Primary Objective, and conversely, all Objectives except the Primary Objective may be flagged as non-critical to the Mission's success. Finally, the Primary Objective is displayed in the Mission Browser as the 'name' of the Mission.

Conjunction

The Mission may depend on the completion one of two Objectives, but not both – therefore, the system must support completing Objective1 OR Objective2 (or both) to determine success. This is an advanced option.

Completion Percentage

The completion of an Objective may be defined as a percentage of success. This is an advanced option.

Types

The verb-action of an Objective may be one of a variety of actions supported by the game system, including, but not limited to Capture, Defense, Delivery, Deploy,

Escort, Destroy, Recon, or Retrieval. More complex actions might include specific Skill use specified by the objective. This feature has advanced options.

Targets or Cargo

The noun-target of an Objective may define a specific Item, *Han's Blaster*, or a non-specific Item Type, *any Blaster*. The target may be a character, item, structure, vehicle, and nearly every other game object. How this interface is accomplished is still in design. Some form of clipboard may allow players to drag and drop representative Items or Item Types into Mission Objectives, and may create 'snapshots' of the game object to use as the target.

Quantity

The Mission Creator may define a quantity for the Objective's target, such as 100 Dewback Hides. By default this value is one (1). Optionally, a minimum or maximum quantity may also be specified when the Mission is created. The Objective conditions are met if the minimum quantity is provided. This latter feature is an advanced option.

(Q: Can we define reward amounts on a per unit delivered basis?)

Destinations

Most Mission Objectives will also have a Destination, which in turn may be any PC, NPC, structure, vehicle, 'chunk egg', or arbitrary region defined by the Mission Creator.

Completion Restrictions & Subcontracting

The Mission Creator may assign Completion Restrictions to each Objective allowing the Mission Leader to subcontract that Objective to another character. This feature provides a method to subcontract portions of larger Missions to characters not capable of handling the whole Mission by themselves. For instance, a manufacturer accepts a recurring Mission to provide an Association with Blasters. After crafting the Blasters at nominal cost, the manufacturer subcontracts the actual delivery to another character, paying them a fraction of the original Mission reward. This is an advanced feature.

- Acceptor Only
- Acceptor or Subcontractor
- Subcontractor Only
- Subcontractor with Filters

Subcontracting Filters

Completion Restrictions may be further enhanced with optional filtering to only allow subcontracting to other characters in the same Faction, Guild, Town, or Association. This is an advanced feature.

Rewards

The Mission Creator may assign a reward to each Objective. A Reward field is automatically provided by the interface, though its content and value are defined as "None" and zero (0). The default values are not presented to the Mission Leader if left unchanged. Both tangible and intangible items may be placed in the Reward field.

Mission Failure: Penalties

The only penalty that can be placed on a Mission for failure is a Bounty. Players may filter Missions that have the Bounty flag. Players have the option of setting a flag to NEVER accept missions with a Bounty. The Mission Creator must fund the Bounty. If a Mission with Bounty is failed, the Mission Creator is notified when they are online and given the option to accept or forgive the Bounty. They do not have to respond immediately and will be notified of the Bounty every time they log in for XX real days. Before accepting a Bounty, the Mission Creator may put credits and items into the Bounty Reward. Once accepted, a Bounty Hunting Mission is generated and posted to the Mission System.

Mission Linking

Mission Creators must be able to link to other missions (within a Mission Package) based on the success or failure of any Objective, or upon the completion state of the Mission itself. This is an advanced feature.

Active Mission Status

Multiple characters can accept some Missions; therefore, for each character allowed to accept the Mission, the Mission's status, Objective completion status, Mission Recurrence status, and subcontracting data must be stored separately for each Mission Leader character.

Mission Leader Status (stored per character)

- Mission List
  - Mission ID
  - Mission Start Time
  - Mission Recurrence Completed Quantity
  - Objective List
    - Objective ID
    - Objective Subcontractor
    - Objective Subcontractor Mission ID
    - Objective Status
      - Objective Completed Quantity

Mission Packages

A Mission Package is a collection of Missions associated by branching and/or linking as defined by a Mission or Objectives therein. A Mission Package may be submitted to the Mission System and only the first Mission in the series, as specified by the creator, will be posted to the Mission Board. Other Missions from the Mission Package are posted as Objectives or Missions are either completed or failed.

Some of the data required for a Mission Package includes the following:

Mission Package

- Mission Package ID
- Mission Package Creator ID
- Mission Package Name
- Mission List
  - Mission ID
- Initial Mission ID

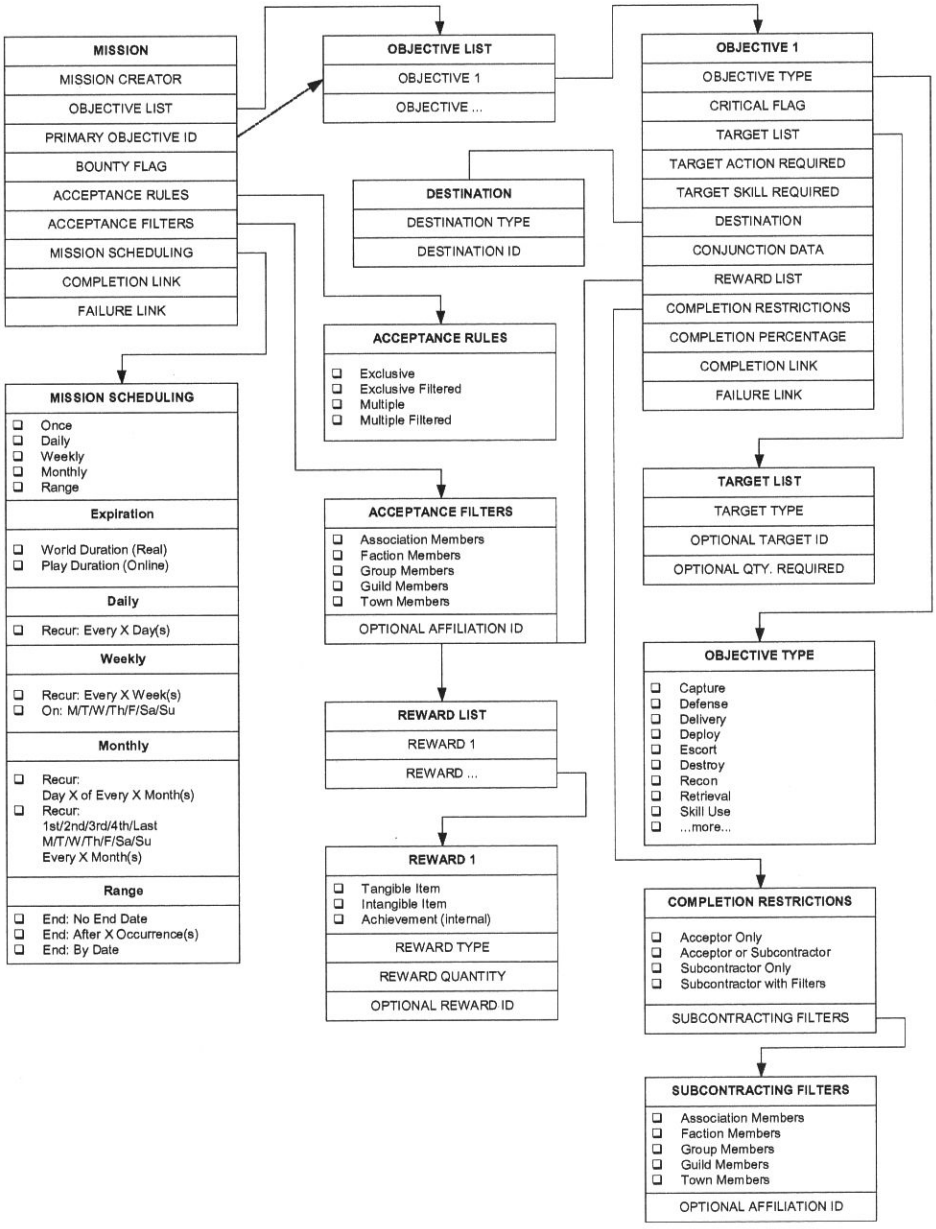
Mission Elements

The Mission Elements Diagram below provides an overview of some of the data required to create the various Mission types.

Some data are editable from within the many advanced options in the Mission Editor. Other data are hidden from players entirely, used internally by game systems for Mission handling. An overview of the elements listed below can be found earlier in this document.

More data elements than shown may be required to properly construct Missions under the current requirements.

Mission Elements (Diagram)



Browsing Missions

How does one browse missions?

Data Terminals

Missions are browsed on a character's *datapad* interface, through a direct interface with a public or private data terminal. To avoid character queues at public terminals, all characters may use data terminals simultaneously. Data terminals are both a ship upgrade and a municipal upgrade.

Guild Terminals

Special data terminals, such as those placed in Guild structures, pre-filter mission types to coincide with the Guild type. For instance, a Bounty Hunter frequenting her Guildhall data terminal to browse missions will always see Bounty Hunting missions first. All other missions are available by quickly changing the filter. Players with their own data terminals may pre-configure the mission filters.

IDEA: Non-critical information can only be viewed at data terminals, thus reducing the more continuous load on the database, and encouraging users to meet other players where public data terminals exist.

Localization

A primary goal for the Mission Browser is to localize as much text as possible. A comments section may be provided for each Mission, allowing a player to enter text in their native language; this text would be non-localized, unsorted, and probably not translated unless we can acquire translation software.

Filtering & Sorting

When browsing at a data terminal, Missions are pre-filtered and sorted based on the default configuration of the terminal. Other criteria may hide Missions from a character. Only members can view and accept Missions created for their specific player Association, Faction, Group, Guild, or Town. If the Affiliation Type is defined but the specific affiliation name is left undefined, a member of that type of affiliation may view and accept the mission.

Players may filter and sort by several criteria such as Mission Type, Reward Value, Acceptor Filters, Bounty, etc.

Accepting Missions

How does one take a mission?

Players accept Missions by browsing to a Mission appropriate for their character and accepting the Mission through the Mission Browser interface. Another method to accept a Mission is through direct interaction between two players where a Contract is formed. An accepted Mission is by nature considered a Contract.

Completing Missions

How does one complete a mission?

Fulfilling all of the critical Objectives defined by the Mission completes a Mission. The methods for fulfilling Objectives include:

- Capture of a Target
- Defense of a Target
- Delivery of a Target to a Destination
- Deployment of a Target
- Destruction of a Target
- Reconnaissance of a Target
- Retrieval of a Target, possibly from a specified Destination
- Skill use (successful) on a Target
- ...And many more.

Mission Packages & Linking

How does one create a Mission Package?

A Mission Package is created through an interface provided by a *datapad* and later uploaded to the Mission System through a data terminal. One or more (undefined at this time) Mission Packages may be stored on a *datapad*. Elements used by a Mission are entered using a 'snapshot' clipboard system provided by the *datapad*.

How are missions linked, to create larger plot threads?

Missions are linked to other Missions by ID, either through Objectives that are completed or failed, or through the completion or failure of the current Mission. Only Missions within a Mission Package can be linked together. Two Missions (A & B) may link to the same Mission (C), for instance, allowing the failure of either Mission A or Mission B to link to Mission C. The Mission Linking system will not allow recursion, however.

Subcontracting Mission Objectives

How do subcontracts work?

Only individual Missions Objectives flagged to allow subcontracting might be subcontracted to other characters. The subcontracting data includes additional information to filter which characters may be subcontractors. The Mission Objectives interface for accepted Missions must provide a way to subcontract Objectives flagged for that action.

The act of subcontracting generates a new Mission on the Mission Board that is associated with the primary (parent) Mission. The Mission Leader of the primary Mission may define whatever Reward they wish for the subcontracted Objective. When the newly generated Mission has been completed, the subcontractor's Reward is paid and the Objective within the primary Mission is flagged as complete. If the subcontractor fails, the primary Mission Leader is notified. They may also be given the opportunity to attempt to subcontract again, but only if the primary Mission has not expired or the Objective Target is still available.

Completion Metrics

What sort of completion metrics do we track for ranking players?

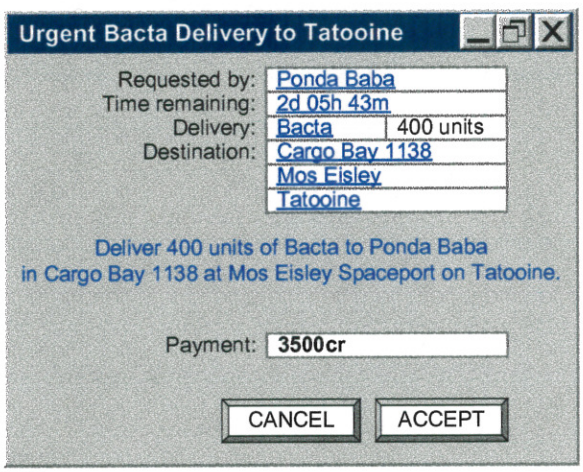
Missions Attempted/Completed
Missions Attempted/Completed By Mission Type (includes subcontracts)

Mission Fees Earned/Lost (credits)
Mission Fees Earned/Lost By Mission Type (credits)
Missions Subcontracted Out Attempted/Completed
Missions By Creator (Last 10, no duplicates, none from affiliations)
Missions Created
Missions Created By Type
Missions Created Attempted/Completed

Core Requirements

- ❑ Mission Packages & Linking
- ❑ Missions with multiple Objectives
- ❑ Mission Browsing with Filtering and Sorting
- ❑ Mission Acceptance with Filtering and Prerequisites
- ❑ Mission Subcontracting with Filtering and Prerequisites
- ❑ Mission Failure Penalty in the form of Bounties
- ❑ Mission Expiration and Recurrence
- ❑ Mission Exclusivity Settings
- ❑ Mission Completion Metrics
- ❑ Rewards per Objective
- ❑ Objective Completion Percentage Requirements
- ❑ Objective Conjunction
- ❑ Mission Linking per Objective
- ❑ Critical and Non-critical Objectives

Interface



TBD

Assets Required

TBD

Revision Log

2.	18-Oct-00	Draft revision

AI-Generated Missions

The computer will be required to spawn missions as filler content for situations where there is perceived to be a lack of sufficient amounts of such content, and for situations where the player can be given the ability to act upon information that they normally would not have access to. Any missions that the computer generates need to be of at least a modicum level of interest, and they must be built using a system that is not totally brought down by the interference of players outside the mission. Finally, the computer-generated mission system must accomplish all of this without its products feeling 'mathematicy and mad-libby' or repetitive.

Vision

Issues for Newbies

- The game must be accessible to a broad audience.
- The game should offer a sense of discovery.
- It should be possible to play a satisfying game session in less than an hour.

To take advantage of the Computer-Generated-Missions system, no special action is required of the player. All the player has to do is show up, and they will have as many opportunities per unit of time to embark on missions as game-veterans and power-gamers.

The computer will be able to generate missions using content pulled from asset lists that will allow for new gaming experiences that the player may not have experienced before. With some additional player-metric tracking, we can weight mission-creation assets in such a way that we can offer new content to the player as often as we want.

With the computer tracking appropriate player-metrics, the computer can know that a player normally engages in shorter sessions and so can weight toward the creation of both shorter and time-independent mission types and quests.

Live service issues

- We must have an average subscriber lifespan of twelve months or more.
- Backend costs must be kept to no more than 40% of a user's monthly fee.

By having a system for the generation of contextually appropriate yet varied content in unlimited locations we can maintain a customer's interest for a longer period of time.

With the generation system tied to a quest generation system, we can unload low to mid-range quest generation from paid staff and allow the system to do the work.

Technology issues

- It must be easy to extend and add content to the game after ship.

Since the majority of the content for the mission generation system is pulled from asset lists, we can quite simply add to the asset lists and impact multiple systems throughout the game, including computer generated missions.

Retention

- *Star Wars Online* must provide multiple ladders of advancement so that new ways to play the game become apparent over time.
- No one profession, specialty, or skill will be optimal—we will use "rock-paper-scissors" techniques to balance differing capabilities.
- There will be motives and rewards for short session play.
- We will explicitly reward non-advancement based ways of playing the game.

Not all missions generated are hack-n-slash missions, with some involving the locating, transportation, or creation of items, in addition to other actions. These varied mission objectives will reward multiple types of skill-sets. Combat-oriented personnel will be valuable, but won't have the market cornered when it comes to completing computer-generated missions and quests.

By using player-metric tracking it will be possible to make some planning for the kind of missions that particular players prefer, and if a player is only ever on for short periods of time, computer generated missions made available to them will include whole missions or quests with sub-sections that are readily complete-able in a short period of time. While some missions will involve the use of skills that are based on use when it comes to advancement, many of them will concern reputation advancement, factional advancement, or outright greed.

Gameplay issues

- The game must satisfy each of Bartle's Four: Achievers, Killers, Socializers, and Explorers.
- We will scatter in as many little touches that add depth as we can. Wherever possible, these touches will relate to hidden or interesting aspects of the *Star Wars* universe.
- There must be a sense of exploration.

The mission generation system is capable of spawning missions and quests with a wide variety of objectives that appeal to multiple types of characters and players.

By pre-building some mission encounter-templates and objectives we can help bring players face-to-face with situations and characters that have significance from the movies; searching for and trading with a band of Jawas crawling over Tatooine's surface in a sandcrawler, for example.

As players roam planets, cities and space at random, they will encounter objects, characters, and creatures and be asked to go on missions or quests. By tracking a character's mission history and maintaining a significant asset list that is guided by types of territory and local events, a player is constantly encountering new things and asked to go places they haven't been before.

Terms

SW: AED – Star Wars: An Empire Divided. The name of this game.

System Details

Core Requirements

- Mission template fulfillment
- Asset list access
- Reliance upon in-game objects and the spawning sub-system

In order for the mission system to automatically generate any missions at all, the server must be able to construct them by selecting and completing the same mission templates that will be used in the creation of player-created missions, and then post them where players will have access to them.

When creating a mission, the computer will choose a mission goal and use this goal to narrow its selection in choosing a specific mission-type template, such as a transport item or destroy item template. After template selection, the computer fills in the blanks on the template using the same categories of asset lists used to fill in the blanks by players. These blanks list off the involved objects and parties, as well as objective locations and the like. A major difference between player and computer generated missions is that the computer has access to the entire spectrum of items from each category available to it when it fills in a blank. The player might only have access to objects, characters, and locations they have seen before, or generic versions of those objects and locations. The assets involved with the mission might be pulled from the pre-existing game-object population, but will also draw upon the SW: AED's object spawning system to fill in content where needed.

After the mission has been generated the computer posts the mission on the same mission assignment boards for posting player-generated missions. No effort is made to differentiate a computer-generated mission from a player-generated mission.

To manage a mission, the computer creates 'tokens' that may sometimes manifest themselves as objects to help in tracking and controlling computer-controlled missions. A player that selects a computer-generated mission would have any new mission-specific content spawned relative to the player (where applicable), so as to minimize the amount of time the mission-specific content is floating around the game-scape and able to be messed with by other players. In situations where the player is supposed to acquire something from a mission-specific character, the required data or object will usually be spawned as a 'token'-object in the possession of the spawned character and flagged as indestructible for the duration of that leg of the mission. If an uninvolved player frags the mission-specific character that a different player was trying to contact or trade with, the content that the player needed isn't destroyed. The token might be on the ground, on the dead character's corpse, or in the possession of the player that fragged the character. If the 'token' item is acquired by the player charged with the mission before the mission's expiration date elapses, then they can continue to complete the mission as normal.

With these capabilities, a rudimentary computer-generated mission system can function enough to buff up the player mission boards during droughts, as well as attaching missions to some randomly spawned NPC encounters. Missions generated and used for these purposes will be primarily of entertainment value only, although with the system in place, external systems can call upon the computer mission generation system to create a mission to fulfill a particular need, with required parameters supplied by the mission creation request. That way, missions required for character advancement or game story purposes can be generated to serve the game mechanic needs.

Quest structure

Quests and mission packages are groupings of missions that go together and are to be completed in a particular order. They can range from being just a couple of missions long to being ten or more missions long. These packages of missions can be assembled in two ways, with the primary difference between the two being their presentation to the player. One method is called 'containment', while the other is called 'linked'.

The containment method of mission assembly involves a set of missions with a stated final objective, and a listing of all the sub-objectives or missions that need to be completed for the overall objective to be considered successful. When a player pulls up a containment-based mission, they know everything that will be asked of them before they can attain the final goal. It's all laid out for them, including multiple paths to victory, if any.

A 'linked' method of mission assembly only displays one mission of a set at a time. Once that portion of the overall objective is completed, the next part is displayed, and the player is given the option of 'opting-out' and deciding not to continue. The player only sees one slice of the mission-set at a time and is never sure of exactly how long it is. If a player played intermediate mission, they would see the briefing and data for the next mission come up, and given the option to abort and not attempt it if they so desire. Occasionally, there is a possibility that there might be penalties for opting-out, but generally the option is present so that a player can choose not to proceed with a mission that is longer or more hazardous than they expected, or requires that the player achieve something unachievable for them.

Linear and branching quests

Both quest structure types come in linear and branching varieties, with some mixing between types also being encountered. In a linear quest, each mission comes after a previous one, and if any mission is lost, the chain is broken and the quest is over. A branching quest involves at least one and sometimes several alternate branches. Sometimes these are 'losing branches' which are chains of missions that are followed if the player doesn't achieve every single of the missions objectives laid out before them in the previous mission. Some quests will branch early in their series, run linear for a bit, then branch again or perhaps recombine back into the original mission chain.

One example of a branching quest structure might involve Luke Skywalker getting from the wampa cave, to Echo base, and escaping in his X-Wing. In the first mission, he must escape the Wampa cave. If he gets out of the cave without being wounded at all, he engages AT-AT's in a snowspeeder in his next mission. If he

was wounded, then he spends some time in a bacta tank and spends the next mission running through Echo base fighting Storm Troopers while trying to get to his X-Wing. As long as the Luke survives either mission, he arrives at the final mission where he must launch in his X-Wing and make it past the blockade to a point where he can hyperspace out of the system.

Computer generated mission and quest availability

Encounter links

There are a couple of different locations that a player might come into contact with a computer-generated mission or quest.

The most simplistic would be on the mission board where all the other missions are listed. If the mission board goes through a slow period where less than what we deem a critical number of missions are being posted, then the computer begins to generate missions until the mission-board population reaches a minimum point. Beyond just tracking the raw number of missions available, the computer will be aware of the percentages of varying types of objectives listed on the board, and can spawn new missions to fill any voids in available mission-types depending on how reactive we choose to make the generation system.

A more interesting and more complicated way of latching onto a computer generated mission or quest would be to encounter it will out and about the game world. Random encounters, both hostile and random, will sometimes have characters or 'tokens' to which missions are attached. By dealing with a game-selected character or picking a 'token' item off a derelict vehicle or body, the player is presented with the opportunity to follow up on a mission or set of missions. Once the item to which the missions were attached had been secured, the player would have the option to follow up on the mission, or to ignore it. Sometimes, missions that were generated for the player that they have chosen not to pursue might be placed on the mission board, or passed on to another player when the token is transferred.

Features & Future Requirements

While the basic computer mission-generation system is quite useful and valuable, it's value and effectiveness can be multiplied many-fold by adding some additional capabilities and functionality to it. To really make computer-generated missions an integral part of the game, this sub-system will need the ability to track and act upon changes in mission and player population metrics. Making the system more dynamic will make it far more compelling.

Customizing missions for the player and world

Several systems in the game require the tracking of player-associated 'metrics' or criteria. By having the mission-generation system also track player histories and capabilities as well as some server statistics and world events, we can increase the power of the system. By knowing what kind of missions the player usually chooses, or what kind of skills or equipment they have, the mission generation system can be weighted in favor of the chances of some objectives being used, or perhaps more widely scattered locations used instead of local areas. If the player lacks or has some critical equipment, certain mission objectives might be totally locked out, or enabled. Also, the system can

more tightly control mission deadlines by measuring the player's ability to travel and making sure that they have enough time to reach the location of the mission objective.

If the player spends only short amounts of time playing each day, then we can weight the mission system towards creating missions that are quicker to accomplish by making more of the mission objectives and requirements local, easier to find, and less complicated to defeat or create. We can also generate missions or quests that are extremely time-tolerant or perhaps not time-dependant at all..

If a player has tons of combat skills and very few crafting skills, we can weight generation in favor combat-oriented missions. If the player has tons of crafting skills, then we can skew mission creation toward missions involving item repair, modification, or creation.

If a player has their own landspeeder, then we can slightly increase chances that they will need to travel in order to accomplish the mission. If a player has their own spaceship, then we can increase chances that they will have to recover something in space or move an item or personnel between planets. If the player has neither vehicles, nor spacecraft, nor ticket-money, then we can decrease chances that anything of that nature might be required. If the player is loaded-down with heavy weaponry, then the player might be asked to assault a base or engage some enemy armored vehicles. If the player has no such gear, but does have the ability to use it, then maybe the system would increase the chances that such a weapon might be made available to them as a mission reward, or an item on a corpse or locked in a bases' locker..

If there are areas of the map that the player has never been to before, then they can be offered missions that take place in those locations, while at the same time checking to see that they have the will and means to get there, such as a vehicle, or a history of traveling.

If in each of the last three missions the player encountered Tusken Raiders riding on banthas, then we can reduce the chances that they are encountered again. If they ARE encountered again, then we can go the next step and lock out Tusken Raiders, banthas, or both from the player's next mission or quest.

Outside of the player-related metrics, if the mission system is given access to world/chunk currently active population numbers, we can weight the chance that mission objectives will take place in areas that are not commonly frequented. This allows us to help get more value out of the existing game terrain in terms of using under-utilized areas, and also help control server loads. It also helps cut down on the opportunities for individuals operating independently of the mission to interfere with it, since there would be fewer bystanders to muddle things up.

To help control the game's item-population, computer mission generation can be slowed down, or set to utilize unused or non-player-owned objects already in the game. This sort of new-item spawn restriction can controlled like a spigot, allowing it be ramped up or down, or even shut off, if game mechanics warrant it. This will hamper mission generation, but the system should be robust enough that such a situation would not leave computer mission generation completely hamstrung.

Another example of a game-world metric would be a major event like a planetary invasion. If there exists an on-going planetary invasion then missions with

objectives involving combat and relief missions increase. You might be actually involved in fighting, or running guns, or transporting food, or healing the wounded, or smuggling people out of harm's way.

The power of the computer mission generation system is increased many times if we allow it to track some information about the current world situation, and who it is building the mission for.

Spawned encounter memory function Tie-in

The spawning system will see considerable use and have a significant affect on how the SW: AED plays. There will be a few times when rather unique characters and equipment are spawned, or spawned hostiles engage and destroy many people. An advanced capability of the spawning system is to save off these 'special' encounters in order to allow people to encounter them again.

If such a saved-encounter listing is available, then the mission/quest generation system can be given the option of utilizing it. For example, players that have frequented the Jundland wastes might spin yarns of a crazed group of bezerker Tusken Raiders that have learned to ride patched-up swoop speeder-bikes; the player's mission is to find their encampment and destroy the bikes. Alternatively, some players might speak of an insane bantha that went crazy in town one day and totaled three landspeeders before escaping into the desert; the insurance company doesn't want to pay up and the player has been charged to bring back the psycho bantha as proof of what happened.

Such missions are far more compelling than something semi-randomly rolled-up off the computer's encounter and mission tables, since they involve notable and sometimes spectacular events that other players are probably talking about.

Service call-ups

Players may elect to be members of organizations that are sometimes called into action to deal with special situations. Although the player might not normally be aware of what is happening in a different location, a 'alert' can be sent out over the SW: AED's ICQ system and request the player's action in some way or another. While some of these situations might be spawned and pre-generated from scratch, the system is even more compelling when it is used to 'react' to current game events. When an event occurs that is flagged as a trigger for an alert, a mission is immediately spawned with all the mission objectives being filled in using the triggering event's participating game objects and players. This mission event is then flashed out to all members of the agencies that would be responsible for dealing with the situation. Using the ability for the mission system to consider the player's capabilities and location, these alerts would only be sent to players that were capable of responding to them. If something happens to make the mission objectives unachievable before alert-responders have been on-scene for a reasonable period of time, the objectives for the alert-mission can be updated or even rescinded with no penalties to the players involved.

The system can be imagined working a little like volunteer fire fighters and their pagers. This game mechanic can be used to bring players into contact with one another in both competitive and cooperative conditions.

If a player or group of characters got together and jumped an Imperial Postal Service delivery ship in Correllian orbit, any players that had signed on to be members of the 'Correllian Civil Space Militia' or 'CorSec Search and Rescue' would have an alert

come across their messaging system requesting assistance with defense and recovery operations relating to the ongoing attack. When the alert comes up, the player can choose to deny it, or they can accept the mission and become subject to its objectives.

If an Imperial base comes under attack by rebel forces, then all imperial-reserve personnel in the area are immediately alerted and requested to come to the base's aid. In some of these situations, players in a military reserve might be loaned military gear to use in the action, then required to return it after the immediate situation was resolved. While a player might never be able to purchase or steal an AT-ST, they might be assigned one temporarily if they are a reservist and are answering a call to active duty.

Players that have signed up to be members of the ready-reserve fleet might be alerted that a plague has broken out on Kessel and medicines are desperately needed. An alert would be sent out to all available reservists, asking them to pick up medical supplies from Naboo and take them to Kessel immediately.

Maybe a group of rebel fighter pilots that were alerted to deal with a Star Destroyer that jumped into Naboo orbit are getting hammered. An alert is sent out to all declared rebels in the area with access to tools and with crafting skills. They are asked to report to a particular base and help repair the fighters as they return to refuel and rearm. If any players accept, an additional alert is sent to the fighter pilots telling them that repair is available at a particular base, with the amount of repair mentioned as being available (slight, large-scale) is relative to how many rebel crafters responded to the call, and how skilled they are.

Players receive an income when they respond to alerts, in addition to sometimes being able to collect rewards. If players deny an alert request, there may or may not be any penalties depending on how many they have ignored previously. A player's history of successful alert responses can be rewarded with 'ranks' within the organization that give bonuses in access to equipment and services. A player with a long history of successfully responding to rebel alerts as an infantryman could be given a pool of points that could be used to requisition a better rifle and armor for fighting with. A player in the 'Tatooine Red Cross' that uses his own landspeeder for running emergency supplies to towns might be able to spend a pool of points on repairs, fuel, or upgrades.

At the same time we are working to bring players together, we can also help control the number of them that are in a particular area. By sending out a limited number of alerts while compensating for alert denials and thereby controlling the size of the local population, the system helps to limit server loads.

Properly implemented, such a 'call-up' system could be incredibly compelling.

Interface

The parts of computer-generated missions that will require player interaction will use components and windows standardized by other parts of the mission system, and other systems in the game. Mission information windows, confirmation windows and the like will all be found in other areas of the game, and as a result the computer-generated missions will not require any new interface.

Assets Required

No unique art is required for the computer-generated mission sub-system. It borrows art from other systems such as player-missions,

Revision Log

Rev 1.0 – Initial generation of document.
Rev 1.1 – Initial polish pass completed. Minor sub-paragraph on tokens added.

Player-Generated Missions

Overview

The mission system (and the contract system) provides players with a means for creating binding agreements. It allows players to decide upon a course of action and to determine the consequences of the possible outcomes. In order to ensure that the system is flexible enough to cover a wide variety of agreements, we must define as many uses of the system as possible. We must then make sure that all of the conditions for use are traceable by the system.

This document will outline both the various uses of player-generated missions and the abstract ways in which players interact with the mission system. It will not try to specify the interface exactly.

Vision

- It should have an intuitive user interface. If the choice for initial interface is between flexibility and ease of use, we will choose ease of use, and make the flexibility an option.
  - Interface is extremely important here, I just wanted to emphasize it ☺
- Players should feel empowered to take action against bad apples in the player base. They should never feel just like victims.
  - The mission system's reward/penalty system gives players some modicum of control over their experiences with other players.
- Human administrative intervention must be minimized.
  - Having a system for handling agreements and player interactions should lower the incidents of intervention because of scamming and misbehavior.
- We must have full support for localization, including Unicode support on client and server.
  - Definitely a requirement for the mission system.
- New strategies should come to light for the player as their play time continues.
  - Although available to newbies, the mission system should really come into it's own for more advanced players.
- *Star Wars Online* must provide multiple ladders of advancement so that new ways to play the game become apparent over time.
  - Mission leader boards, mission completion metrics…
- Player-driven content will be a key element of the game.
  - Player-generated missions! What else can I say?
- There will be motives and rewards for short session play.
  - The mission boards are perfect for the player who wants to jump in, take a mission, complete it, and go.
- The game must satisfy each of Bartle's Four: Achievers, Killers, Socializers, and Explorers.

- Mission boards and history for achievers, missions for killers, controlled player interactions for socializers…3 out of 4 isn't bad... ☺
- ❑ We must encourage and support a community.
  - The more you can trust other players, the easier it is to "commune"
- ❑ Players need tools to shape their own experiences.
  - Player-generated missions!
- ❑ There should be ladders and leader boards for everything in the game, including social activities.
  - Mission Type leader boards!
- ❑ There must be a wide range of in-game communication systems.
- ❑ Robust support for guilds and other such player organizations is a requirement.
  - Player Associations can generate and accept missions

Terms

Player Contract – A formal binding agreement between players. Its conditions are enforced by the system. Contracts are created through the player's personal datapad.

Datapad – The player's interface for a variety of systems, including the mission system.

Datapad Object Memory – The system that players use to keep references to objects for use with the mission system. The datapad is the player's interface to this system.

Dataterminal – An object in the game that allows a player to access specific mission boards. Used to post and accept missions.

Player Mission – A player mission is a contract that is posted for acceptance by a remote party. The difference between a contract and a mission is that with a contract, both parties are present for the agreement. Player missions can be found on different mission boards that are accessed via Dataterminals.

Details

Core Requirements

For details on the underlying mission system, please refer to missions system document.

Player-generated missions come in two forms:

Player Missions: A player publishes a mission using a Dataterminal. Example: Ja'Qwan has 6 containers of Wookie pelts he needs to ship from Tatooine to Naboo. Using the Datapad, he's able to create a new mission specifying what needs to be transported, from where, to where, how much he's willing to pay, etc… He then goes to the local commodities market to upload his mission at the Dataterminal.

Player Contracts: A player uses his or her datapad to create a contract between his or herself and another party. Example: Ja'Qwan meets a player named Burundi who says he'll take him to Hoth for 100Cr. Ja'Qwan brings up his datapad and quickly creates a contract that says Burundi will take Ja'Qwan to Hoth in return for 100Cr, and that failure will result in a 50Cr bounty on Burundi's head. Burundi receives the contract and can choose to accept or reject it.

The Datapad Object Memory (DOM)

Both player missions and player contracts use the Datapad Object Memory (DOM) system. The DOM is a collection of pointers to real objects in the world. Any time a player encounters an object in the game, they can store a pointer to that object in the DOM. Then, when they go to create a mission, they can bring up the DOM and use the entries to create missions. To the player the DOM is nothing more than a list of names of objects and their types. If objects have intrinsic visual representations (icons) then we may also choose to use these in the DOM to make it easier to navigate. Otherwise the DOM will be a text list of items. For example, it may appear as a spreadsheet with:

"name," "type," "date stored"
Ja'Qwan, Player, 10/17/2000
Oola, NPC, 8/23/2002

Perhaps players can click on the item name and get more detail. The DOM probably needs its own overview document. Items in the DOM can be given to another player at anytime. The interface should support DOM transactions with remote players (i.e. dragging an item from the DOM onto someone in your friends list). As with most player interactions, players need to be able to specify who can give them DOM data (DOM /ignore functions for example).

The DOM has a LIMITED INVENTORY. Players can only have so many items in the DOM at a time. Also, since the items in the DOM are essentially references to object id's in the game, it probably needs to clean up occasionally. We also need to look into problems with ID recycling. At this time, Chris Mayer and Justin Randall assure me that this is not an issue.

The Player Mission Editor (PME)

Although the method for instigating missions and contracts differ, the mechanics of setting up the guts of the agreement are not different. As a matter of fact, the only difference between the two is that the "Mission Leader" field is automatically propagated in a Player Contract (the target of the contract) and there is no option for the Player Contract creator to "accept".

When a player begins a new contract or mission, they enter the Datapad Player Mission Editor (PME). PME Functionality:

- The PME allows the player to propagate the appropriate fields of a mission including mission type, objectives, limitations, rewards, penalties, etc...
    - For any field that accepts an "object" as an argument, the PME provides a list of all known objects to the player. This list includes any possession of the player (personal inventory, player venue, player venue inventory, ship, vehical, AND any entry in the DOM.

- The PME allows the player to save a Mission as a "template". Templates are saved client side.
- The PME allows the player to create a Mission or Contract from a template.

Generating Player Missions

For every type of player mission, there is an associated "mission board" that can only be accessed via the appropriate dataterminal. Examples of player mission boards include: **The Commodities Market** (shipping transport and shipping escort missions), **The Assassins' Mission Board** (The assassin overview hasn't been written yet, but only certain players can be assassinated, usually they hold a particular position that they will lose if assassinated), or any **Player Association's Internal Mission Board.**

In order to create a mission for one of these boards, a player must have access to the appropriate Dataterminal. Dataterminal mission publishing access may be restricted by any combination of the following:

- Licensing - See commodities market overview
- Location – To access a dataterminal, a player must be able to physically reach it. It may be decided that a player only has to be within a certain range, or in the same structure as the dataterminal (player venues for example).
- Group Membership – To post a mission on the "Bob's Badass Bobafett Brotherhood" player association mission board, not only do you have to gain access to the terminal, you must also be a member of the PA. We can even consider implementing a permission system within the PA that grants posting privileges to its members.
- Number of currently published missions. Players should be limited in the number of missions they can post. The specific number of concurrent missions will be determined at a later date and may differ per type of mission/mission board.

Datapads and Dataterminals

The abstract mechanism for generating a player mission is the use of the Datapad. Once a mission is generated in the datapad, the mission can then be "uploaded" from the Datapad to the Dataterminal when the player accesses the terminal. The purpose for this is to allow players to generate missions ahead of time without having to be near a dataterminal. Some missions, particularly packages, may be fairly complex and it would be useful for the player to be able to work on them wherever they are.

"Uploading" isn't a coincidental term. To allow players the ability to create as many missions as they desire, missions will be stored on the **client** until they are published via the dataterminal. Is there an offline editor? Probably not, the PME needs to have access to the data in the Datapad Object Memory, which is probably only available when the player is connected.

Accessing Player Missions

The method for accessing available player generated missions is similar to that of publishing. It requires the player to access an appropriate Dataterminal. Different Dataterminals provide access to different mission types. For example, the public Commodities Market Dataterminal will not allow players to access the Assassins' Guild Mission Boards.

The method for accessing a player's current missions (and perhaps their status?) is the player's datapad. At any time, a player can use their datapad to see what missions they currently have and what their mission completion stats are.

Creating Player Contracts

Player Contracts are agreements generated directly between players. The reason for them to exist is so that players can interact with others on their own terms.

For example, suppose J'Qwan meets Burundi in a Player-run bar, and Burundi says "Hey, I have a new spaceship, let's go to Hoth and hunt some Taun-tauns!" J'Qwan doesn't know Burundi very well, and is afraid that Burundi may strand him on Hoth. So he says, "Sure! Lemme pull up a quick contract." He accesses his Datapad, and whips up a quick contract that essentially says "Burundi will take J'Qwan to Hoth and then return J'Qwan to Hoth, or J'Qwan will put a 100Cr bounty on Burundi's head." If Burundi agrees, J'Qwan at least SOME leverage in the relationship. If J'Qwan and Burundi continue to hang out, are they likely to continue to use the Player Contract this way? Probably not, but this is only one example. Anytime two people who have no trust base, or are risking a great deal, they'll have the contract system to shore-up their interactions.

The interface should make contract generation *EASY*. It would be fantastic if anytime a player interacts with another player, they could create a contract . Player Contracts require interaction controls, just like the DOM. Players should be able to prevent other players from creating contracts with them (to avoid spam). For a variety of reasons (storage, complexity), there should be a limitation on the number of outstanding contracts a player can have at any given time.

Leader Boards and Reputation

The mission system should keep track of all player mission metrics, including how many missions they've completed/failed and of what type. This information should be accessible by other players (especially those offering missions) when necessary. We should also have leader boards for the different types of missions (shipping, escort, transportation, assassination, bounty hunting, etc). Ideally, these metrics should include who generated the mission. The leader boards should take that information into consideration when generating the rankings (taking multiple missions from the same account or player association should lower the effect of a mission completion).

Features & Future Requirements

It would be fantastic to see a system of advancement in the faction system that allowed high-ranking faction members to create military missions!